

**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE QUITO**

**CARRERA:**

**INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de**

**INGENIERO ELECTRÓNICO**

**TEMA:**

**IMPLEMENTACIÓN DE UN CONTROLADOR PID DIFUSO PARA  
MANTENER EL EQUILIBRIO EN UN ROBOT BALLBOT DURANTE EL  
DESPLAZAMIENTO**

**AUTOR:**

**JOSÉ LUIS PAREDES LEÓN**

**TUTOR:**

**EDUARDO RODRIGO TORRES SANTOS**

**Quito, julio 2021**

## **CESIÓN DE DERECHOS DE AUTOR**

Yo, José Luis Paredes León con documento de identificación N°1723244586, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación: IMPLEMENTACIÓN DE UN CONTROLADOR PID DIFUSO PARA MANTENER EL EQUILIBRIO EN UN ROBOT BALLBOT DURANTE EL DESPLAZAMIENTO, mismo que ha sido desarrollado para optar por el título de Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



-----  
José Luis Paredes León

CI. 1723244586

Quito, julio 2021

## **DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR**

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, IMPLEMENTACIÓN DE UN CONTROLADOR PID DIFUSO PARA MANTENER EL EQUILIBRIO EN UN ROBOT BALLBOT DURANTE EL DESPLAZAMIENTO realizado por José Luis Paredes León, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, julio 2021



-----  
Eduardo Rodrigo Torres Santos.

C.I. 1713654604

## **DEDICATORIA**

Este logro va dedicado a mis padres, gracias por siempre brindarme su apoyo incondicional, a pesar de las adversidades de la vida.

Valoro mucho su esfuerzo, y las necesidades que tuvieron que pasar, para darme la oportunidad de tener una carrera universitaria, y por ende una mejor forma de vida,

Gracias padres por su cariño y confianza, por sus noches de desvelo, por su preocupación y enseñanzas, que ha sido eje fundamental para culminar esta meta, que se las dedico a ustedes, con todo mi corazón y espero con este logro y con el tiempo compensar todo lo que han hecho por mí.

## **AGRADECIMIENTOS**

Este logro no hubiese sido posible sin la bendición, y la luz de mi Dios que siempre ilumino mi camino por más oscuro que se encontrase.

Enseñándome que mientras haya vida, hay fe.

Y esa fe nos llevara a coronar las cubres más difíciles.

## ÍNDICE GENERAL

PORTADA.....	i
CESIÓN DE DERECHOS DE AUTOR.....	ii
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR.....	iii
DEDICATORIA .....	iv
AGRADECIMIENTOS .....	v
RESUMEN.....	xi
ABSTRACT.....	xii
INTRODUCCIÓN .....	xiii
ANTECEDENTES.....	1
1.1    PLANTEAMIENTO DEL PROBLEMA.....	1
1.2    JUSTIFICACIÓN.....	1
1.3    OBJETIVOS.....	2
1.3.1    Objetivo General .....	2
1.3.2    Objetivos Específicos.....	2
1.4    METODOLOGÍA.....	2
1.5    ALCANCES .....	3
MARCO TEÓRICO.....	4
2.1    ANTECEDENTES DEL BALLBOT.....	4
2.2    CONCEPTOS GENERALES DEL ROBOT .....	5
2.2.1    Cuerpo en equilibrio.....	5
2.2.2    Robot ballbot.....	5
2.2.3    Modelo dinámico .....	6
2.3    SISTEMAS DE COORDENADAS .....	7
2.3.1    Rotación de ejes coordenados .....	7
2.3.2    Ángulos de Euler.....	8
2.4    CONTROLADORES PID.....	9
2.5    REGLAS DE ZIEGLER NICHOLS .....	10
2.5.1    Curva con forma de S.....	10
2.5.2    Método de la ganancia máxima.....	11
2.6    LÓGICA DIFUSA.....	12
2.6.1    Conjuntos difusos.....	12

2.6.2	Control difuso.....	12
2.6.3	Control de Sintonización PID difuso .....	13
DISEÑO E IMPLEMENTACIÓN.....		14
3.1	SELECCIÓN DEL BALLBOT.....	14
3.2	BALLBOT EV3 .....	16
3.2.1	Bloque EV3.....	16
3.2.2	Sensor de giro.....	16
3.2.3	Servomotor.....	17
3.2.4	Conexiones del bloque EV3.....	17
3.2.5	Implementación del Ballbot EV3.....	18
3.3	MODELADO DEL ROBOT.....	19
3.3.1	Descripción del modelo.....	19
3.3.2	Diseño cinemático.....	20
3.3.3	Diseño dinámico del sistema.....	22
3.4	DISEÑO DE CONTROLADORES PID.....	24
3.4.1	Función de transferencia del motor reductor .....	25
3.4.2	Sintonización PID mediante Ziegler Nichols.....	28
3.4.3	Diseño del controlador PID difuso.....	31
3.5	SOFTWARE EV3 .....	35
PRUEBAS Y RESULTADOS.....		38
4.1	PRUEBAS DE EQUILIBRIO CON PID ZIEGLER NICHOLS .....	38
4.2	PRUEBAS DE EQUILIBRIO CON EL CONTROLADOR PID DIFUSO	40
4.2.1	Prueba inicial.....	40
4.2.2	Calibración final.....	42
4.2.3	Comparación entre los 2 PID difusos.....	44
4.3	PRUEBAS Y RESULTADOS ENTRE LOS 2 CONTROLADORES PID	46
4.3.2	Ángulos de giro .....	47
4.3.3	Pruebas con diferentes niveles de peso .....	47
4.4	PRUEBAS DE DESPLAZAMIENTO CON EL PID DIFUSO.....	48
4.5	CONCLUSIONES.....	50
4.6	RECOMENDACIONES .....	51
REFERENCIAS.....		52
ANEXOS.....		54

## ÍNDICE DE FIGURAS

Figura 2.1. Ballbots a) CMU, b) TGU, c) y d) Lego Mindstorms .....	6
Figura 2.2. Modelo Ballbot 2D .....	7
Figura 2.3. Rotación en el plano $xy$ .....	7
Figura 2.4. Secuencia de rotación alrededor de los ejes a) Z, b) Y, c) X.....	8
Figura 2.5. Bloques de control PID.....	10
Figura 2.6. Respuesta en forma de S.....	10
Figura 2.7. Oscilación sostenida .....	11
Figura 2.8. Conjuntos a) convexos y b) no convexos .....	12
Figura 2.9. Componentes del control difuso .....	13
Figura 2.10. Bloque PID difuso .....	13
Figura 3.1. Kit de Lego Mindstorms.....	15
Figura 3.2. Lego Mindstorms a) RCX, b) NXT y c) EV3 .....	15
Figura 3.3. Parte frontal del bloque EV3 .....	16
Figura 3.4. Sensor de giro EV3 .....	17
Figura 3.5. Servomotor EV3 .....	17
Figura 3.6. Puertos y parlante del bloque EV3 .....	18
Figura 3.7. Ballbot Evolution EV3 .....	19
Figura 3.8. Dimensiones de las omni-ruedas .....	19
Figura 3.9. Modelado del robot.....	20
Figura 3.10. Prisma rectangular .....	23
Figura 3.11. Función de transferencia total.....	24
Figura 3.12. Circuito eléctrico y mecánico del motor reductor .....	25
Figura 3.13. Función de transferencia del motor reductor .....	28
Figura 3.14. Sistema del robot en lazo cerrado .....	29
Figura 3.15. Respuesta del sistema en lazo cerrado con $K_p = 100$ .....	29
Figura 3.16. Respuesta del sistema en lazo cerrado con $K_p = 113$ .....	30
Figura 3.17. Respuesta del sistema en lazo cerrado con $K_p = 112.3$ .....	30
Figura 3.18. Diagrama de bloques del sistema con PID Tuner.....	32
Figura 3.19. Respuesta del sistema con el PID Tuner.....	32
Figura 3.20. Funciones a) pertenencia (entradas) y b) membresía (salida) .....	33
Figura 3.21. Función de pertenencia difusa .....	35



Figura 3.22. Conexión para la configuración MicroPython.....	35
Figura 3.23. MicroPython en Visual Studio Code .....	35
Figura 3.24. Encabezado principal del programa EV3brick .....	36
Figura 3.25. Movimientos del Ballbot EV3 .....	36
Figura 3.26. Cálculo de la salida del control PID difuso .....	37
Figura 4.1. Posición de los sensores de giro .....	38
Figura 4.2. Comparación de los ángulos de giro con PID Ziegler Nichols .....	39
Figura 4.3. Giro vs Voltaje 1 con Ziegler Nichols .....	40
Figura 4.4. Giro vs Voltaje 2 con Ziegler Nichols .....	40
Figura 4.5. Comparación del ángulo de giro con PID difuso inicial.....	41
Figura 4.6. Giro vs Voltaje 1 con PID difuso inicial .....	42
Figura 4.7. Giro vs Voltaje 2 con PID difuso inicial .....	42
Figura 4.8. Comparación de giro con el PID difuso final .....	43
Figura 4.9. Giro vs Voltaje 1 con PID difuso final .....	44
Figura 4.10. Giro vs Voltaje 2 con PID difuso final .....	44
Figura 4.11. Comparación de constantes del controlador PID 1 .....	45
Figura 4.12. Comparación de los ángulos de giro con el PID difuso.....	46
Figura 4.13. Comparación de constantes PID Ziegler Nichols vs PID difuso .....	46
Figura 4.14. Ángulos de giro con PID Ziegler Nichols y PID difuso .....	47
Figura 4.15. Pruebas de funcionamiento del Ballbot EV3.....	48
Figura 4.16. Conexión entre el control remoto y Ballbot EV3 .....	49
Figura 4.17. Desplazamiento del robot en el sentido izquierda derecha.....	49
Figura 4.18. Desplazamiento del robot en el sentido adelante-atrás.....	50

## ÍNDICE DE TABLAS

Tabla 2.1. Parámetros de la curva de respuesta en forma de S .....	11
Tabla 2.2. Parámetros basados en la ganancia crítica .....	11
Tabla 3.1. Robots Ballbot comerciales .....	14
Tabla 3.2. Características del Lego Mindstorms RCX .....	15
Tabla 3.3. Conexión del sensor de giro y servomotores al bloque EV3 .....	18
Tabla 3.4. Parámetros del Ballbot EV3.....	24
Tabla 4.1. Ángulo de giro y voltaje con PID Ziegler Nichols .....	39
Tabla 4.2. Mediciones del giro y voltaje con el PID difuso inicial.....	41
Tabla 4.3. Mediciones del giro y voltaje con el PID difuso final .....	43
Tabla 4.4. Comparación entre constantes PID .....	45
Tabla 4.5. Comparación entre ángulos de giro con PID difuso .....	45
Tabla 4.6. Constantes PID Ziegler Nichols vs PID difuso.....	46
Tabla 4.7. Ángulos de giro con PID Ziegler Nichols vs PID difuso.....	47
Tabla 4.8. Equilibrio del Ballbot EV3 .....	48

## RESUMEN

El presente proyecto consiste en la implementación del controlador PID difuso en el robot Ballbot EV3, este contiene sensores, actuadores, cuerpo EV3, cables de conexión, conectores y emplea el uso del software MicroPython. Partiendo del modelado del robot, se aplica las transformadas de Lagrange en las ecuaciones de movimiento para determinar la aceleración, por ende, se calcula la posición en la salida y es comparada con el dato de entrada (referencia de posición ideal de cero grados) en el sistema de lazo cerrado; como consecuencia se genera un error en la posición por lo que se aplica un control PID difuso en el voltaje de los motores de corriente continua de caja reductora. Cabe mencionar que la parte difusa del controlador PID hace uso de ocho reglas lingüísticas, procesos de fuzzificación y defuzzificación que ayudan a determinar un conjunto idóneo de comandos para que el robot que mantenga el equilibrio durante su desplazamiento sin problemas y desviaciones mínimas. Por consiguiente, se realizan pruebas de funcionamiento con los controladores PID difuso y PID de Ziegler Nichols basado en el método de la ganancia crítica, utilizando diferentes niveles de peso hasta un máximo de 200 gramos sobre su plataforma, en un plano inclinado de tres grados máximo con respecto a la horizontal.

**Palabras clave:** ballbot, PID difuso, robot, controlador para el desplazamiento, Ziegler Nichols

## **ABSTRACT**

This project consists of the implementation of the fuzzy PID controller in the Ballbot EV3 robot, it contains sensors, actuators, EV3 body, connection cables, connectors and uses the MicroPython software. Starting from the robot modeling, the Lagrange transforms are applied in the equations of motion to determine the acceleration, therefore, the position at the output is calculated and compared with the input data (ideal position reference of zero degrees) at the closed loop system; therefore, an error is generated in the position, so a fuzzy PID control is applied to the voltage of the gearbox direct current motors. It is worth mentioning that the fuzzy part of the PID controller makes use of eight linguistic rules, fuzzification and defuzzification processes that help determine an ideal set of commands for the robot to maintain balance during its movement without problems and minimal deviations. Accordingly, functional tests are performed with Ziegler Nichols fuzzy PID and PID controllers based on the critical gain method, using different weight levels up to a maximum of 200 grams on their platform, on a maximum three-degree incline plane with respect to the horizontal.

**Keywords:** ballbot, fuzzy PID, robot, controller for displacement, Ziegler Nichols

## INTRODUCCIÓN

El Ballbot es un robot móvil cuya función es permanecer estable sobre una bola, es decir una rueda esférica gracias a su estabilidad dinámica. Cada día, se ha ido estableciendo diferentes códigos de programación para mantener su equilibrio dependiendo de su peso, forma, altura y su entorno, trabajando de manera omnidireccional en planos rectos. El controlador proporcional integral derivativo (PID) difuso es exclusivo para el control del Ballbot con diferentes pesos y en un plano inclinado.

El presente proyecto se desarrolla en cuatro partes, enumeradas a continuación:

- El Capítulo uno abarca lo relacionado al planteamiento del problema, justificación, objetivos y metodología del proyecto.
- En el Capítulo dos se describe el marco teórico donde contempla lo relacionado al robot ballbot, cuerpo en equilibrio, sistema de coordenadas, modelo dinámico, controladores PID y lógica difusa.
- El Capítulo tres contiene la selección e implementación del ballbot, modelado del robot, diseño de controladores PID difuso y Ziegler Nichols; y el uso del software MicroPython.
- En el Capítulo cuatro se realiza el análisis del Ballbot en software y puesta en marcha del robot. Una vez realizado el análisis, se procede con las conclusiones y recomendaciones.

Cabe mencionar que el presente proyecto técnico posee referencias bibliográficas y anexos.

# **CAPÍTULO 1**

## **ANTECEDENTES**

### **1.1 PLANTEAMIENTO DEL PROBLEMA**

Con el pasar de los años se han ido presentando diferentes plataformas de robots dinámicamente estables, cada una con alguna mejoría en función de las anteriores, resolviendo algún problema que se adapte al tiempo y necesidad en la actualidad. Se ha pasado desde bases grandes y pesadas que presentan para garantizar bajos centros de gravedad, hasta robots dinámicamente estables, altos y delgados, que trabajan en planos horizontales o verticales y en una sola dirección, pero ¿hay robots omnidireccionales y que trabajen en planos inclinados?

En esta área se encuentra el robot Ballbot, donde año tras año se han ido estableciendo diferentes códigos de programación para mantener su equilibrio dependiendo de su peso, forma, altura y su entorno; trabajando de manera omnidireccional en planos rectos, pero no en inclinados, controlados con un sistema PID Fuzzy.

El problema del presente proyecto radica en realizar el control de su equilibrio con un PDI difuso y diferentes pesos, moviéndose de manera omnidireccional, en un plano inclinado, capaz de responder a cualquier cambio en su entorno y comparar su comportamiento en un PID sintonizado con Ziegler-Nichols.

### **1.2 JUSTIFICACIÓN**

Tanto en el campo de la robótica móvil como en los sistemas de control automático, la estabilidad dinámica en lazo cerrado es un factor primordial para el control del equilibrio durante el movimiento de los robots móviles. Gracias a ello, han surgido algunas leyes de control cuyo propósito es optimizar los recursos para estabilizar un sistema, además, sea rápido y confiable en el desempeño (García, 2014).

Para realizar dicho sistema se pretende emplear una estructura ballbot para implementar un controlador PID Fuzzy. Al final del proyecto se espera contar con un sistema el cual tenga un comportamiento óptimo en su desempeño comparado con una técnica tradicional del control automático, como es el método de Ziegler-Nichols, que sirva a estudiantes y profesores para llevar a la práctica conceptos teóricos de

ingeniería de control y robótica móvil de una manera fácil e interactiva, empleando una plataforma accesible y polivalente.

Se desarrollará un robot móvil dinámicamente estable, dinámicamente ágil, como para maniobrar fácilmente en planos inclinados y ceder fácilmente cuando es empujado.

### **1.3 OBJETIVOS**

#### **1.3.1 Objetivo General**

Desarrollar un controlador PID difuso para el equilibrio del Ballbot, durante sus desplazamientos.

#### **1.3.2 Objetivos Específicos**

- Caracterizar un robot Ballbot para establecimiento de las variables de entrada y salida.
- Implementar un controlador PID difuso para control del equilibrio del Ballbot.
- Desarrollar un control remoto para darle la dirección de desplazamiento del Ballbot.
- Realizar pruebas con diferentes niveles de peso sobre su plataforma, en un plano inclinado, como máximo 200gr, para analizar el comportamiento de su equilibrio entre un PID sintonizado con Ziegler-Nichols y el PID difuso.

### **1.4 METODOLOGÍA**

Se iniciará con el método de investigación exploratorio de los diferentes tipos, estructuras y costos para el robot Ballbot y así obtener a través del método deductivo el robot más adecuado para la ejecución del artículo. Además, en el método descriptivo se hará una evaluación de las características del robot, para describir sus partes, comportamiento y o estado de las variables. Por consiguiente, se utiliza el método empírico para el desarrollo del control PID Fuzzy y una investigación cuantitativa junto con el método estadístico de los resultados para determinar la eficiencia del control.

El desarrollo del control remoto se lo hará con el método inductivo, tomando en cuenta las diferentes variables del sistema. Por último, se utiliza el método cuantitativo para

analizar el comportamiento del robot con los dos sistemas de control empleados, es decir, mediante el PID Fuzzy y Ziegler Nichols.

## **1.5 ALCANCES**

El proyecto de titulación se refiere principalmente a los componentes de programación del robot ballbot que será seleccionado de los que se encuentren en el mercado y sea de fácil acceso. Una vez realizado la programación y desarrollo del proyecto, el equipo será entregado a los laboratorios de electrónica de la Universidad Politécnica Salesiana Sede Quito.



## **CAPÍTULO 2**

### **MARCO TEÓRICO**

En la actualidad, la relevancia e importancia de los robots ha aumentado de forma significativa a los hogares, tales como: cortadoras de césped, robots de limpieza, drones con cámara y juguetes. De esta manera, se estima que en los próximos años los robots faciliten la vida diaria y cambien la forma de interacción humana con su entorno. Pues bien, el uso de los robots educativos se orienta a los estudiantes cuyo propósito tiene como fin promover un espíritu emprendedor, despierta el interés en las diferentes áreas de conocimiento, creatividad y experimentación. En el caso de los robots móviles que utilizan plataformas de bajo costo como Adept, Moway, Epuck y Lego Mindstorms, no ofrecen una alta precisión como los robots industriales, pero son válidas para actividades educativas, investigaciones y pruebas experimentales (Lepuschitz & Merdan, 2019).

#### **2.1 ANTECEDENTES DEL BALLBOT**

A continuación, se presenta una breve descripción de ballbots desarrollados en los últimos 5 años:

Wang et al. (2019) desarrollaron el artículo Modeling and controlling of spherical omnidirectional por el Instituto de Ciencia y Tecnología del Norte de China, el cual dispone de un sistema PID en cascada para realizar un control sobre el movimiento omnidireccional y equilibrio del robot. Además, emplea modelos para determinar los parámetros internos de control del motor, establece la parte dinámica con la ayuda de la herramienta SimMechanics y por medio del Simulink se desarrolla la simulación del algoritmo de control. En consecuencia, durante las pruebas experimentales se obtuvo un control eficaz y robusto ya que la desviación en la dirección real del robot está dentro de los 2 grados, inferior a lo indicado en la variación del ángulo de Euler que es de 3 grados.

Johnson et al. (2018) desarrollaron el artículo A fuzzy logic-in-loop control for a novel reduced height ballbot por el Instituto de Ciencia y Tecnología de Kattankulathur India. A diferencia de los robots tradicionales, este fue diseñado con una mínima altura a fin de que el centro de gravedad sea cercano al centroide de la esfera y no provoque esfuerzos adicionales en el controlador. La distribución de las ruedas

omnidireccionales se encuentra a una separación de 120 grados, tal que se fijan a través de un eje para que el plano de las ruedas sea perpendicular al plano tangente. Además, el recorrido del ballbot utiliza ecuaciones cinemáticas de avance, estas son expresadas con matrices y utiliza sensores ópticos para detectar la dirección del movimiento de la bola. La velocidad del motor paso a paso se realiza por medio de un factor proporcional y la simulación del modelo cinemático fue a través del Simulink de Matlab. Por último, utiliza emplea la lógica difusa como la sustitución de las incertidumbres y cálculos del modelo de cinemática inversa, el cual asigna una entrada con tres variables de salida por medio de reglas if-then; este cálculo se basa en progresiones las cuales son ingresadas al microcontrolador.

Malý (2018) desarrolló el tema de tesis Design and realization of the ball-riding robot for the promotion of the faculty por la Universidad Técnica Checa de Praga, el cual se observaron las siguientes actividades: presenta modelaciones y control de movimiento del Ballbot, este robot se ha desarrollado mediante el Lego Mindstorm EV3 con sus 3 omni-ruedas y una bola de boliche para el equilibrio, utiliza las ecuaciones de Lagrange en el modelo dinámico del robot al movimiento sobre un plano y emplea el regulador cuadrático lineal denominado en inglés como linear quadratic regulator (LQR) para la estabilización del punto en equilibrio.

## **2.2 CONCEPTOS GENERALES DEL ROBOT**

### **2.2.1 Cuerpo en equilibrio**

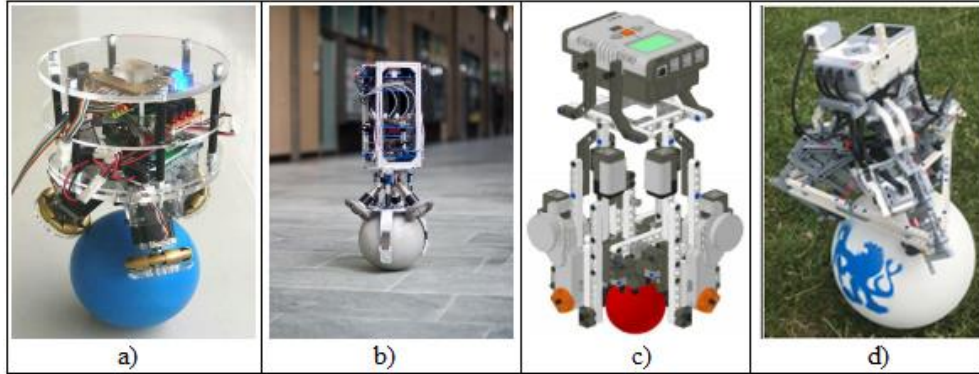
Un cuerpo se encuentra en equilibrio cuando permanece en reposo o si la velocidad es constante, es decir, si el cuerpo posee un equilibrio estático o dinámico, respectivamente. Cuando el cuerpo se encuentra en equilibrio, se aplica la primera Ley de Newton, donde el sumatorio de fuerzas o la resultante sobre la partícula debe ser matemáticamente igual a cero, esta es una condición necesaria y suficiente para el equilibrio (Pérez H. , 2015).

### **2.2.2 Robot ballbot**

El ballbot es un robot cuya función es accionar una rueda esférica y mantenerla en equilibrio gracias al uso de las teorías de control y al concepto de estabilidad dinámica, sin depender de la gravedad. Además, utiliza varias omni-ruedas por lo que representa

ser un robot ágil, confiable, permite una inclinación espontánea en cualquier dirección y gira sobre su propio eje (Chengtao, Jiaxin, & Zuoyong, 2019).

Figura 2.1. Ballbots a) CMU, b) TGU, c) y d) Lego Mindstorms



Fuente: (Malý, 2018)

Por lo general, las partes físicas del robot constan de un cuerpo, sensores, motores y la rueda esférica. En las Figuras 2.1.a) y 2.1.b) se observan prototipos ballbots desarrollados por las Universidades de Carnegie Mellon y Adelaide, respectivamente. En cambio, las Figuras 2.1.c) y 2.1.d) son ballbots desarrollados mediante el Lego Mindstorms NXT y EV3, respectivamente (Malý, 2018).

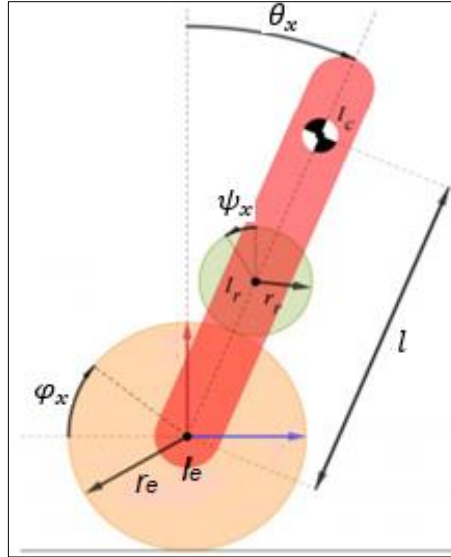
### 2.2.3 Modelo dinámico

La dinámica en un cuerpo rígido comprende la composición de los movimientos de traslación y rotación (Pérez H. , 2015). El Ballbot requiere ecuaciones en movimiento del modelo dinámico, estas se determinan mediante las derivadas de las ecuaciones. Para ello, se maneja el modelo 2D visto en la Figura 2.2 que representa una simplificación del modelo en 3D (Cruz & Gutiérrez, 2019).

Donde:

- $\theta_x, \psi_x$  y  $\varphi_x$ : representa el ángulo de desplazamiento robot, rueda virtual y pelota; respectivamente.
- $I_c, I_r$  e  $I_e$ : es el momento de inercia del cuerpo del robot, rueda virtual y pelota; respectivamente.
- $r_r$  y  $r_e$ : representa el radio de la rueda virtual y esfera, respectivamente.
- $l$  es la distancia medida desde los centros de masas: robot y esfera.

Figura 2.2. Modelo Ballbot 2D



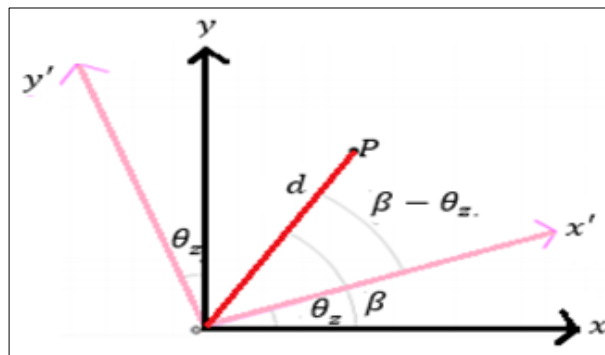
Fuente: (Cruz & Gutiérrez, 2019)

## 2.3 SISTEMAS DE COORDENADAS

### 2.3.1 Rotación de ejes coordenados

La rotación consiste en hacer girar en cierto ángulo con respecto al eje perteneciente. En la Figura 2.3 se observa la rotación de un ángulo  $\theta_z$  respecto al eje  $z'$ , eso implica que el marco de referencia  $\{x', y', z'\}$  ha rotado con respecto al marco  $\{x, y, z\}$ . Evaluando al marco referencial rotado, se tiene una rotación tanto en el eje  $x$  como el eje  $y$ . De tal manera, que existe un punto  $P$  ubicado a una distancia  $d$  del origen de ambos marcos de referencia (Quesada, 2017).

Figura 2.3. Rotación en el plano  $xy$



Fuente: (Quesada, 2017)

En coordenadas polares, el punto  $P$  se expresa mediante la Ecuación 2.1 con respecto al par de coordenadas  $xy$ , donde  $d\cos(\beta)$  y  $d\sin(\beta)$  son las posiciones del punto  $P$  y

$\beta$  representa el ángulo que existe entre el segmento  $d$  con respecto al eje  $x$  (Quesada, 2017).

$$\begin{cases} x = d\cos(\beta) \\ y = d\sin(\beta) \end{cases} \quad \text{Ecuación 2.1}$$

En la Ecuación 2.2 se observa el par de coordenadas con respecto al marco de referencia  $x'y'$ , donde  $\theta_z$  representa el ángulo entre el eje  $x'$  y  $x$  (Quesada, 2017).

$$\begin{cases} x' = d\cos(\beta - \theta_z) = d\cos(\beta)\cos(\theta_z) + d\sin(\beta)\sin(\theta_z) \\ y' = d\sin(\beta - \theta_z) = d\sin(\beta)\cos(\theta_z) - d\cos(\beta)\sin(\theta_z) \end{cases} \quad \text{Ecuación 2.2}$$

Por consiguiente, este sistema puede ser expresado como matriz cuadrada, tal como se indica en la Ecuación 2.3.

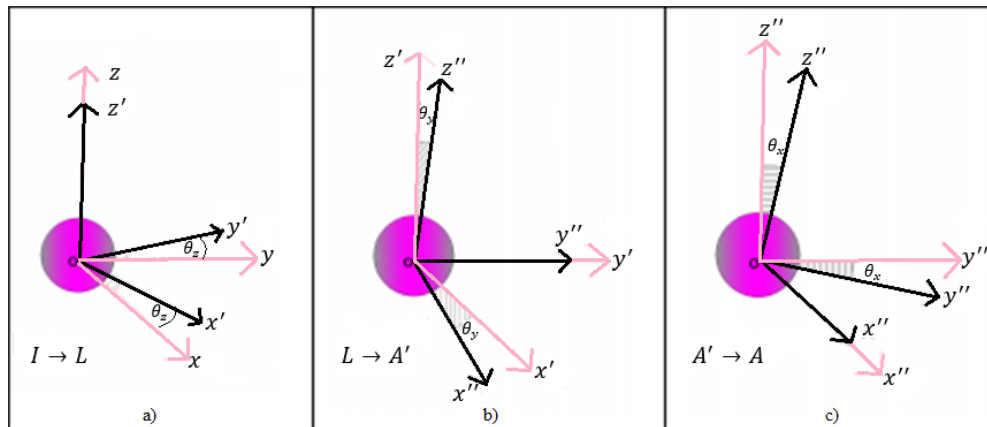
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(\theta_z) & \sin(\theta_z) & 0 \\ -\sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_{x'x} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \text{Ecuación 2.3}$$

Finalmente puede los términos de la matriz cuadrada puede ser representado  $R_{x'x}$  que expresa la matriz de rotación del sistema de coordenadas  $x$  al sistema  $x'$ .

### 2.3.2 Ángulos de Euler

Los ángulos de Euler representan la orientación del sistema de referencia relativo con respecto a otro, donde la rotación se detalla mediante los tres ejes coordenados ortogonales. En la Figura 2.4 se indican las secuencias de rotación alrededor de los ejes  $\{z, y, x\}$  del sistema inercial  $I$  hasta el sistema referencial  $A$  (Quesada, 2017).

Figura 2.4. Secuencia de rotación alrededor de los ejes a) Z, b) Y, c) X



Fuente: (Quesada, 2017)

El desplazamiento se realiza a través de matrices de rotación, de tal manera que la matriz  $R_{IA}$  se observa en la Ecuación 2.4 (Quesada, 2017).

$$R_{IA} = \begin{bmatrix} \cos(\theta_y)\cos(\theta_z) & IA_{1x2} & IA_{1x3} \\ \cos(\theta_y)\sen(\theta_z) & IA_{2x2} & IA_{2x3} \\ -\sen(\theta_y) & \cos(\theta_y)\sen(\theta_x) & \cos(\theta_x)\cos(\theta_y) \end{bmatrix} \quad \text{Ecuación 2.4}$$

Donde:

$$IA_{1x2} = \cos(\theta_z)\sen(\theta_x)\sen(\theta_y) - \cos(\theta_x)\sen(\theta_z).$$

$$IA_{1x3} = \sen(\theta_x)\sen(\theta_z) + \cos(\theta_x)\cos(\theta_z)\sen(\theta_y).$$

$$IA_{2x2} = \cos(\theta_x)\cos(\theta_z) + \sen(\theta_x)\sen(\theta_y)\sen(\theta_z).$$

$$IA_{2x3} = \cos(\theta_x)\sen(\theta_y)\sen(\theta_z) - \cos(\theta_z)\sen(\theta_x).$$

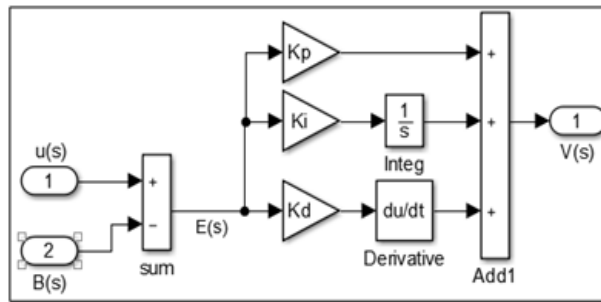
## 2.4 CONTROLADORES PID

Los procesos pueden ser identificados mediante un sistema de control, este a su vez presenta características en el error de estado estable y/o régimen transitorio. Por ello, se añade al sistema un elemento de control cuyo propósito es corregir las características de respuesta de los elementos que se encuentran en la retroalimentación o trayectoria directa. El controlador PID consiste en una combinación de la parte proporcional  $K_p$ , integral  $T_i$  y derivativa  $T_d$ ; cuyo objetivo es obtener un error en el estado estable nulo y anticipar a través de la parte derivativa un efecto predictivo sobre la salida del proceso (Hernández, 2010).

$$G_c(s) = \frac{V(s)}{E(s)} = K_p \left[ 1 + \frac{1}{T_i s} + T_d s \right] \quad \text{Ecuación 2.5}$$

De la Ecuación 2.5, se observa que el controlador posee, un factor proporcional  $K_p T_d$ , un par de ceros debido a la función cuadrática y el polo ubicado en el origen. Además, puede ser representado mediante un diagrama de bloques, visto en la Figura 2.5 (Hernández, 2010).

Figura 2.5. Bloques de control PID



Fuente: (Hernández, 2010)

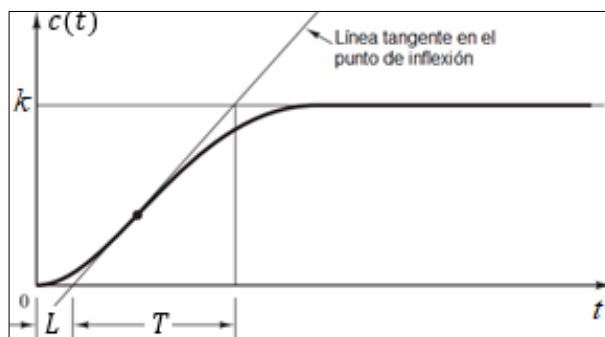
## 2.5 REGLAS DE ZIEGLER NICHOLS

Una vez establecido el método de control, se prosigue con los criterios de sintonización de Ziegler Nichols asignando valores específicos a los componentes del controlador PID. A continuación, se detallan los dos criterios de Ziegler Nichols: curva con forma de S y método basado en la ganancia crítica (Hernández, 2010).

### 2.5.1 Curva con forma de S

Este método se obtiene de forma experimental o a partir de una simulación dinámica ante la respuesta  $c(t)$  a un escalón unitario  $u(t)$  de la planta. Si la función de la planta no contiene polos dominantes complejos conjugados ni integradores, la curva de respuesta tiene forma de S, tal como se indica en la Figura 2.6; esta posee dos parámetros como: tiempo de retardo  $L$  y la constante de tiempo  $T$ . Los parámetros  $L$  y  $T$  se determinan mediante el trazado de una recta tangente en el punto de inflexión de la curva y a su vez se interseca al eje del tiempo (Ogata, 2010).

Figura 2.6. Respuesta en forma de S



Fuente: (Ogata, 2010)

En la Tabla 2.1 se detallan los parámetros  $T$  y  $L$  en función de las constantes del PID.

Tabla 2.1. Parámetros de la curva de respuesta en forma de S

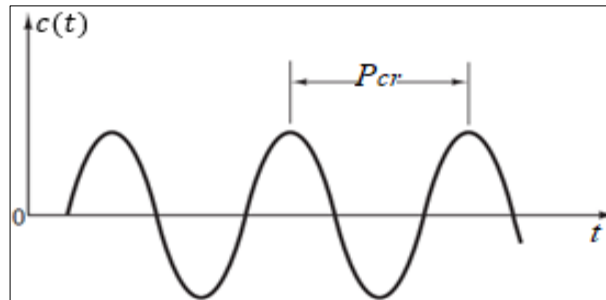
Tipo de controlador	$K_p$	$T_i$	$T_d$
Proporcional P	$T/L$	$\infty$	$0$
Proporcional Integral PI	$0.9T/L$	$L/0.3$	$0$
Proporcional Integral Derivativo PID	$1.2T/L$	$2L$	$0.5L$

Fuente: (Ogata, 2010)

### 2.5.2 Método de la ganancia máxima

El método de la ganancia máxima o también denominado de sintonización en lazo cerrado consiste en el incremento de la ganancia  $K_p$  hasta un valor crítico  $K_{cr}$  a tal punto que la salida del sistema posea una oscilación sostenida, visto en la Figura 2.7 (Ogata, 2010).

Figura 2.7. Oscilación sostenida



Fuente: (Ogata, 2010)

De manera que la ganancia crítica  $K_{cr}$  y periodo crítico  $P_{cr}$  se determinan mediante la Tabla 2.2.

Tabla 2.2. Parámetros basados en la ganancia crítica

Tipo de controlador	$K_p$	$T_i$	$T_d$
Proporcional P	$0.5K_{cr}$	$\infty$	$0$
Proporcional Integral PI	$0.45K_{cr}$	$P_{cr}/1.2$	$0$
Proporcional Integral Derivativo PID	$0.6K_{cr}$	$2P_{cr}$	$0.125P_{cr}$

Fuente: (Ogata, 2010)

Cabe mencionar que  $P_{cr}$  se calcula mediante la frecuencia de oscilaciones sostenidas  $\omega_{cr}$  (Ogata, 2010).



## 2.6 LÓGICA DIFUSA

La lógica difusa se basa en la teoría de conjuntos donde los elementos del universo posean grados intermedios de pertenencia mediante una función característica (Román, 2018).

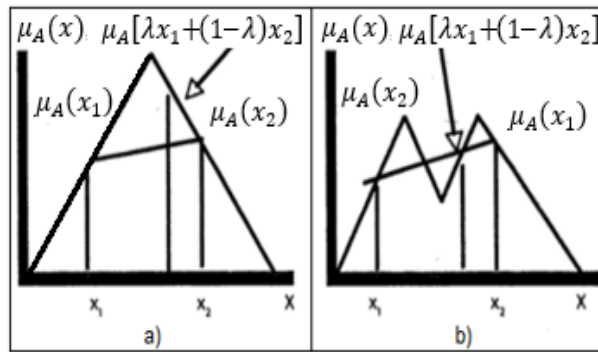
### 2.6.1 Conjuntos difusos

El conjunto difuso posee una función  $\mu_A(x)$ , cuyo valor real está en el rango entre 0 a 1 y representa el grado de pertenencia de  $x$  en el conjunto A. Por lo general, las funciones de pertenencia son de forma cuadráticas, sigmoidales, trapezoidales o triangulares, siendo funciones convexas, visto en la Ecuación 2.6. Por consiguiente, se puede decir que son parecidas a una función de densidad o distribución de probabilidad (Pérez & León, 2007).

$$\mu_A[\lambda x_1 + (1 - \lambda)x_2] \geq \text{Min}[\mu_A(x_1) + \mu_A(x_2)] \quad \text{Ecuación 2.6}$$

Para todo  $x_1$  y  $x_2$  en  $x$  y todo  $\lambda$  dentro del intervalo entre 0 a 1, tal como se observa en la Figura 2.8 (Pérez & León, 2007).

Figura 2.8. Conjuntos a) convexos y b) no convexos

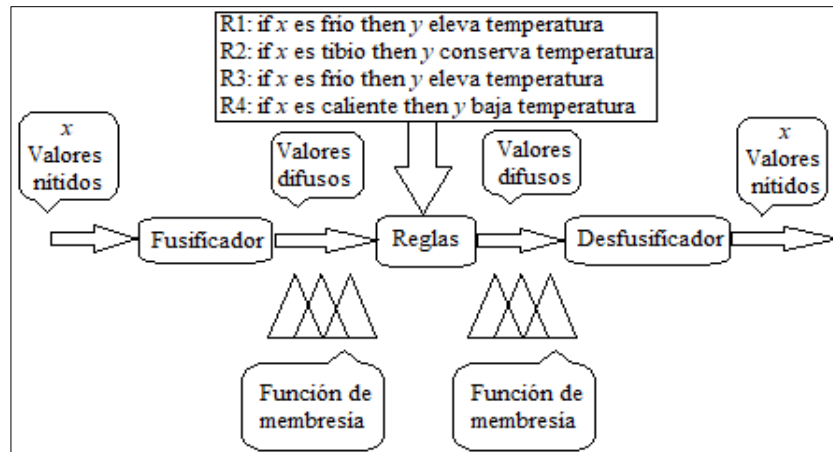


Fuente: (Pérez & León, 2007)

### 2.6.2 Control difuso

El control difuso se compone de tres elementos como: nexo o fusificador que convierte un valor real a difuso de todas las entradas, reglas de la acción de control aplicadas y el desfusificador que es el proceso inverso de la primera parte, es decir, genera una salida real partiendo de las reglas difusas. En la Figura 2.9 se dispone de un control Fuzzy en lazo cerrado utilizando reglas lingüísticas escogiendo como ejemplo la variable de temperatura (Villanes, 2016).

Figura 2.9. Componentes del control difuso



Fuente: (Villanes, 2016)

### 2.6.3 Control de Sintonización PID difuso

Las reglas difusas aplicadas en el controlador PID ayudan a mejorar la sintonización en base a la señal de entrada y derivada del error, tal como se observa en la Figura 2.10. Se ingresan los rangos de acción del robot en base a una matriz de información; por último, se aplica la fuzzificación y defuzzificación (Villanes, 2016).

Figura 2.10. Bloque PID difuso



Fuente: (Villanes, 2016)

## CAPÍTULO 3

### DISEÑO E IMPLEMENTACIÓN

#### 3.1 SELECCIÓN DEL BALLBOT

Uno de los alcances de la investigación cuantitativa se basa en el método exploratorio, estos ayudan a preparar el terreno para examinar el objeto de estudio (Reyes, 2016). Este método se aplica para analizar los diferentes tipos de robots Ballbot comerciales, tal como se muestra en la Tabla 3.1.

Tabla 3.1. Robots Ballbot comerciales

Descripción	Ninco	Lego Mindstorms
Versiones	2	3
Mando inalámbrico del robot	Infrarrojo IR	IR, Bluetooth, USB WiFi
Programación modificable	No	Sí
Alimentación al robot	6 V	9 V
Estructura	Fija	Variable

Fuente: (Ninco, 2020; ElectricBricks, 2020)

En base al método deductivo, se selecciona el Lego de Mindstorms ya que permite el uso de lenguajes de programación adaptables como C, Java, Ada, Python, LabVIEW y Simulink de Matlab; con ello se puede introducir el controlador PID difuso. En cambio, el Ballbot Ninco no permite realizar modificaciones externas de software.

El desempeño de la plataforma de Lego Mindstorms posee serie una de kits referente al hardware (Figura 3.1) y software a fin de crear robots personalizados y programables. Además, posee un bloque inteligente con computadora que controla el sistema, motores modulares, conjunto de sensores y piezas para el ensamblado del sistema mecánico, por lo que es una herramienta constructiva para que estudiantes aprovechen su conocimiento y experiencia para resolver problemas del mundo real (Lepuschitz & Merdan, 2019).

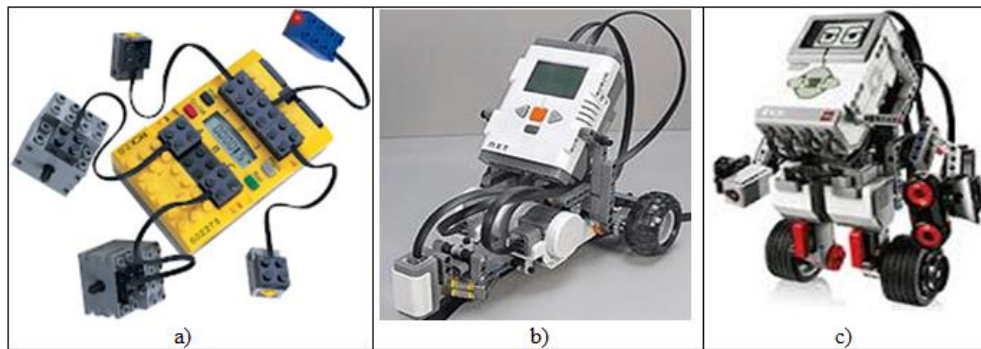
Figura 3.1. Kit de Lego Mindstorms



Fuente: (ElectricBricks, 2020)

En la Figura 3.2 se observan los tres tipos de plataformas que existen en el Lego Mindstorms: RCX, NXT y EV3.

Figura 3.2. Lego Mindstorms a) RCX, b) NXT y c) EV3



Fuente: (ElectricBricks, 2020)

Cada una de las versiones posee ciertas características que difieren con relación a las entradas, salidas, velocidad, procesamiento y comunicación, tal como se indica en la Tabla 3.2. Se selecciona el robot EV3 ya que destaca sobre las demás versiones, por ejemplo: presenta una velocidad de procesamiento de 300 MHz, el cual es 6.25 y 18.75 veces superior al NXT y RCX, en cuanto se refiere a las memorias RAM y FLASH el Ballbot EV3 está en el orden de los megabytes, en cambio el NXT no pasa de los 256 KB. Por último, EV3 dispone de la comunicación WiFi y las dos versiones anteriores no disponen de esta red (ElectricBricks, 2020).

Tabla 3.2. Características del Lego Mindstorms RCX

Características	RCX	NXT	EV3
Entradas	3	4	4
Salidas	3	3	4
Velocidad	16 MHz	48 MHz	300 MHz
Memoria RAM	-	64 KB	64 MB
Memoria Flash	-	256 KB	16 MB
Memoria SRAM	32 KB	-	-
Comunicación	Entre bloques o infrarrojo	USB o Bluetooth	USB, bluetooth, Wi-Fi

Fuente: (ElectricBricks, 2020)

## 3.2 BALLBOT EV3

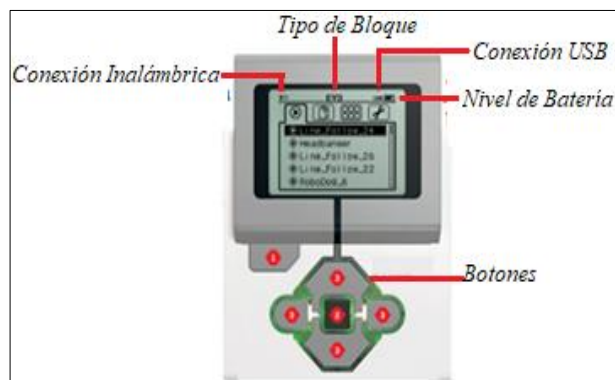
El Lego Mindstorms EV3 posee componentes hardware como: sensores, motores, bloque EV3, ruedas, cables de conexión y fuentes de alimentación.

### 3.2.1 Bloque EV3

El bloque EV3 representa la parte principal del sistema, ya que es el cerebro del robot. En la Figura 3.3 se observa la parte frontal del bloque que contiene una pantalla visual que indica el tipo de bloque, conexión del dispositivo por el puerto USB, el nivel de la batería, la conexión inalámbrica (bluetooth, WiFi) y botones que indican las siguientes acciones (LEGO, 2019):

- El Botón “1” se utiliza para cancelar o revertir acciones del programa y apagar el bloque EV3.
- En cambio, el Botón “2” acepta o verifica las acciones/ajustes del programa.
- Por último, el bloque EV3 dispone de 4 botones denominados “3”, cuya función ayuda a la navegación de los contenidos del bloque ya que pueden desplazarse bajo las 4 direcciones: izquierda, derecha, arriba y abajo.

Figura 3.3. Parte frontal del bloque EV3



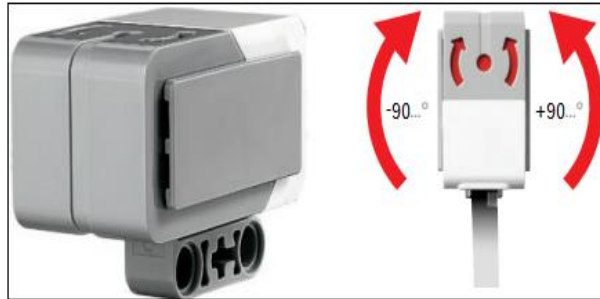
Fuente: (LEGO, 2019)

### 3.2.2 Sensor de giro

Los sensores son dispositivos encargados de obtener información de cierta magnitud física o química mediante una señal eléctrica. Cada sensor ofrece características como rango de medida, precisión, exactitud, entre otras (Corona, Abarca, & Mares, 2019). En la Figura 3.4 se observa al sensor de giro que tiene como objetivo la detección del movimiento de rotación en grados por segundo con exactitud de  $\pm 3^\circ$  en un giro de

90°. Además, el sensor posee unas flechas la razón de la rotación que pueden indicar cuando una parte del robot gira o si el robot se cae (LEGO, 2019).

Figura 3.4. Sensor de giro EV3



Fuente: (LEGO, 2019)

### 3.2.3 Servomotor

El servomotor se conforma de un motor de corriente continua donde el eje se acopla a una caja de engranajes a fin de potenciar el torque del motor y obtener una precisión fija a los movimientos del motor (Corona, Abarca, & Mares, 2019). En la Figura 3.5 se observa el servomotor utilizado para el Ballbot EV3; además, dispone de un sensor de rotación con control de un grado de precisión (LEGO, 2019). Para este proyecto, no se utiliza el sensor de rotación del servomotor, por lo que queda como un motor DC con caja reductora.

Figura 3.5. Servomotor EV3



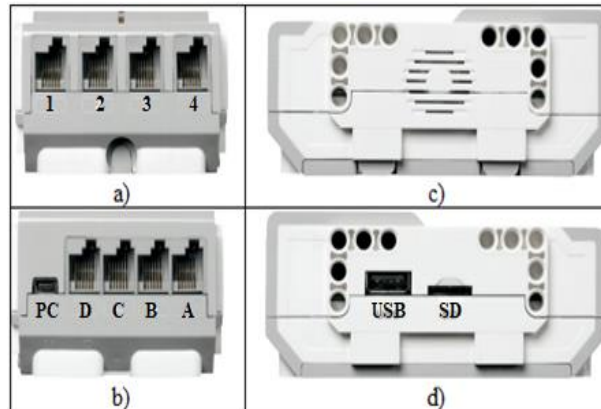
Fuente: (LEGO, 2019)

### 3.2.4 Conexiones del bloque EV3

A los costados del bloque EV3 se dispone de cuatro puertos de entrada para la conexión de los sensores, tal como se indica en la Figura 3.6.a; cuatro puertos de salida para los motores y un minipuerto USB utilizado para la conexión del bloque a un equipo, visto en la Figura 3.6.b. Por consiguiente, se dispone de un parlante (Figura 3.6.c) y en la Figura 3.6.d se observa un puerto USB para establecer una conexión hacia alguna red

inalámbrica o bloques EV3; por último, posee un puerto de tarjeta SD que sirve para aumentar la memoria del bloque hasta un máximo de 32 GB (LEGO, 2019).

Figura 3.6. Puertos y parlante del bloque EV3



Fuente: (LEGO, 2019)

En la Tabla 3.3 se aprecia la conexión del sensor de giro y servomotores en el bloque EV3.

Tabla 3.3. Conexión del sensor de giro y servomotores al bloque EV3

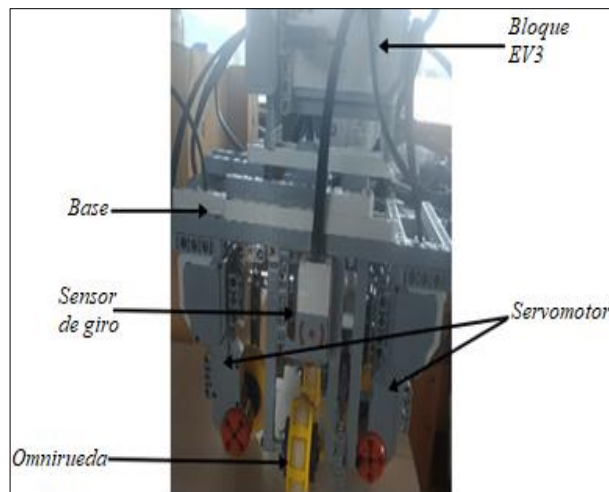
Descripción	Entradas / Salidas	Número del Puerto
Sensor de giro 1	Entrada	1
Sensor de giro 2	Entrada	2
Servomotor 1	Salida	A
Servomotor 2	Salida	B
Servomotor 3	Salida	C

Fuente: Autor

### 3.2.5 Implementación del Ballbot EV3

Mediante el uso de las piezas mecánicas y demás elementos pertenecientes al kit de del Lego Mindstorms se ha implementado el Ballbot EV3. En la Figura 3.7 se observa el bloque EV3 que está ubicado en la parte superior del robot, montando en la parte media gracias al uso de piezas pequeñas adicionales.

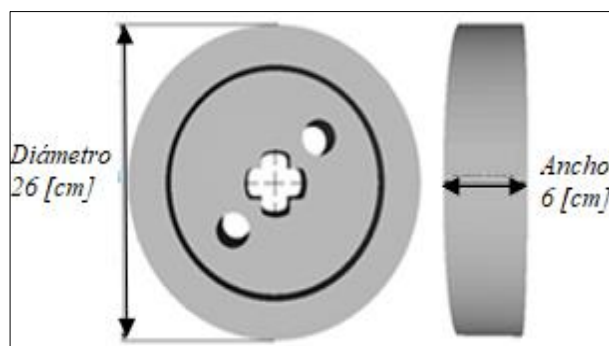
Figura 3.7. Ballbot Evolution EV3



Fuente: Autor

Además, utiliza marcos internos y externos a fin de obtener una mejor distribución de masa y refuerzo de las juntas. Por consiguiente, se coloca en la parte inferior 2 sensores de giro, 3 servomotores y 4 omni-ruedas externas, este último elemento no pertenece a la marca del Lego Mindstorms. El Ballbot presenta dos movimientos en el desplazamiento: atrás-adelante e izquierda-derecha que no operan a la vez entre sí, es decir, si el robot se mueve hacia adelante o atrás, las omni-ruedas izquierda o derecha se quedan estáticas y viceversa; eso genera una mayor fuerza durante el desplazamiento por lo que se considera utilizar omni-ruedas externas de dimensiones 26x6 [cm] (diámetros x ancho), visto en la Figura 3.18.

Figura 3.8. Dimensiones de las omni-ruedas



Fuente: Autor

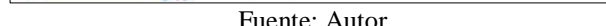
### 3.3 MODELADO DEL ROBOT

#### 3.3.1 Descripción del modelo

El modelo del sistema posee movimientos de rotación y traslación, por lo que se le considera como un sistema de dos grados de libertad. En la Figura 3.9 se observa el modelo representativo del robot Ballbot EV3, cuyo propósito es encontrar las



---



$$x_e' = \varphi_x' r_e$$

$$x_{r_1}' = \varphi_x' r_e + \cos\theta_x (r_r + r_e) \theta_x' + \frac{d\theta_x'}{2} \text{sen}\theta_x$$

$$x_{r_2}' = \varphi_x' r_e + \cos\theta_x (r_r + r_e) \theta_x' - \frac{d\theta_x'}{2} \text{sen}\theta_x$$

$$x_c' = \varphi_x' r_e + l\theta_x' \cos\theta_x$$

Coordenadas en el eje y:

$$y_e = 0 \quad \text{Ecuación 3.5}$$

$$y_{r_1} = (r_r + r_e) \cos(\theta_x - \alpha) \quad \text{Ecuación 3.6}$$

$$y_{r_2} = (r_r + r_e) \cos(\theta_x + \alpha) \quad \text{Ecuación 3.7}$$

$$y_c = l \cos\theta_x \quad \text{Ecuación 3.8}$$

Donde:

$$\alpha = \tan^{-1} \left( \frac{d}{2(r_r + r_e)} \right).$$

La velocidad con respecto al eje y se obtiene con las derivadas de ecuaciones del movimiento:

$$y_e' = 0$$

$$y_{r_1}' = -\theta_x' (r_r + r_e) \text{sen}(\theta_x - \alpha)$$

$$y_{r_2}' = -\theta_x' (r_r + r_e) \text{sen}(\theta_x + \alpha)$$

$$y_c' = -l\theta_x' \text{sen}\theta_x$$

En la Ecuación 3.9, se observa la velocidad de rotación de la rueda virtual  $\psi_x'$  en función de las rotaciones del cuerpo del robot  $\theta_x'$  y la pelota  $\varphi_x'$ , rueda virtual  $r_r$  y rueda de la esfera  $r_e$  (Fankhauser, 2010).

$$\psi_x' = \frac{r_e}{r_r}(\varphi_x' - \theta_x')\theta_x' \quad \text{Ecuación 3.9}$$

### 3.3.3 Diseño dinámico del sistema

Se utiliza las ecuaciones de Lagrange con el propósito de realizar las derivaciones de las ecuaciones de movimiento. Para ello, se determina los siguientes factores (Fankhauser, 2010):

- **Paso 1.** Energía cinética  $T$  de los cuerpos rígidos en función de la cinemática del sistema, tal como se indica en las siguientes Ecuaciones respecto a la bola  $T_e$ , cuerpo del robot  $T_c$ , ruedas  $T_{r1}$  y  $T_{r2}$ . Cabe mencionar que la energía cinética contempla los movimientos de traslación y rotación.

$$T_e = \frac{m_2(\varphi_x' r_e)^2}{2} + \frac{I_e(\varphi_x')^2}{2} \quad \text{Ecuación 3.10}$$

El torque del cuerpo se indica mediante la Ecuación 3.11.

$$T_c = \frac{m_1}{2}[(x'_c)^2 + (y'_c)^2] + \frac{I_c \theta_x'^2}{2} \quad \text{Ecuación 3.11}$$

$I_c$  representa el momento de inercia del cuerpo del robot. Por último, se tiene los torques de las ruedas  $T_{r1}$  y  $T_{r2}$ .

$$T_{r1} = \frac{m_3}{2}[(x'_{r1})^2 + (y'_{r1})^2] + \frac{I_r}{2}(\psi_x')^2 \quad \text{Ecuación 3.12}$$

$$T_{r2} = \frac{m_3}{2}[(x'_{r2})^2 + (y'_{r2})^2] + \frac{I_r}{2}(\psi_x')^2 \quad \text{Ecuación 3.13}$$

Donde:

- $m_1, m_2$  y  $m_3$  son las masas del cuerpo del robot, esfera y rueda virtual; respectivamente.
- $I_r$  indica el momento de inercia de la rueda del Ballbot en el centro.

En la Ecuación 3.14 se aprecia el momento de inercia de la pelota (esfera hueca)  $I_e$ . Por consiguiente, en las Ecuaciones 3.15 y 3.16 se observan las inercias para la rueda

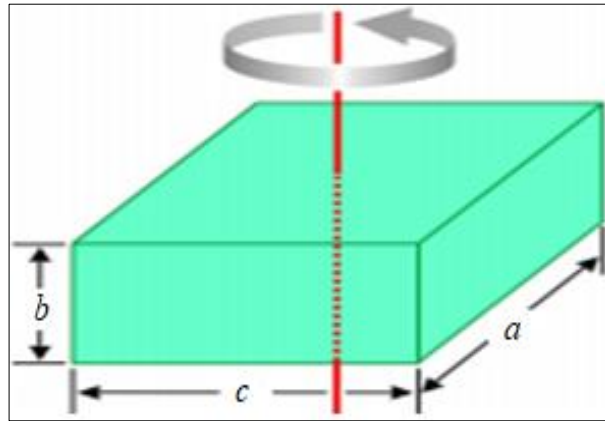
$I_r$  (disco delgado) y cuerpo del robot  $I_c$ , similar a un prisma rectangular, visto en la Figura 3.15 (Vaxasoftware, 2021).

$$I_e = \frac{2m_2 r_e^2}{3} \quad \text{Ecuación 3.14}$$

$$I_r = \frac{m_3 r_r^2}{2} \quad \text{Ecuación 3.15}$$

$$I_c = \frac{m_1 (b^2 + c^2)}{12} \quad \text{Ecuación 3.16}$$

Figura 3.10. Prisma rectangular



Fuente: (Vaxasoftware, 2021)

- **Paso 2.** Energía potencial “V” de los cuerpos rígidos en función de la cinemática del sistema, tal como se indica en la Ecuación 3.17 respecto a la bola  $V_e$ , cuerpo del robot  $V_c$  y rueda virtual  $V_r$ .

$$\begin{cases} V_e = 0 \\ V_c = m_1 g l \cos(\theta_x) \\ V_r = m_3 g (r_e + r_r) \cos(\theta_x) \end{cases} \quad \text{Ecuación 3.17}$$

- **Paso 3.** Establecer el Lagrangiano  $\mathcal{L}$ , tal como se indica en la Ecuación 3.18 (Cruz & Gutiérrez, 2019)

$$\begin{cases} \frac{d}{dt} \left( \frac{\delta T}{\delta \dot{\vec{q}}_i} \right)^T - \left( \frac{\delta T}{\delta \vec{q}_i} \right)^T + \left( \frac{\delta V}{\delta \vec{q}_i} \right)^T = \text{fuerzas externas} \\ \mathcal{L} = T - V \end{cases} \quad \text{Ecuación 3.18}$$

“ $\vec{q}$ ” es el vector de coordenadas generalizadas,  $i=1, \dots, n$ ; donde “n” representa la longitud del vector  $\vec{q}$ .

$$\vec{q} = \begin{bmatrix} \theta_x \\ \varphi_x \end{bmatrix} \quad \text{Ecuación 3.19}$$

En la Tabla 3.4 se exponen los parámetros del sistema, donde todos se rigen bajo el sistema internacional.

Tabla 3.4. Parámetros del Ballbot EV3

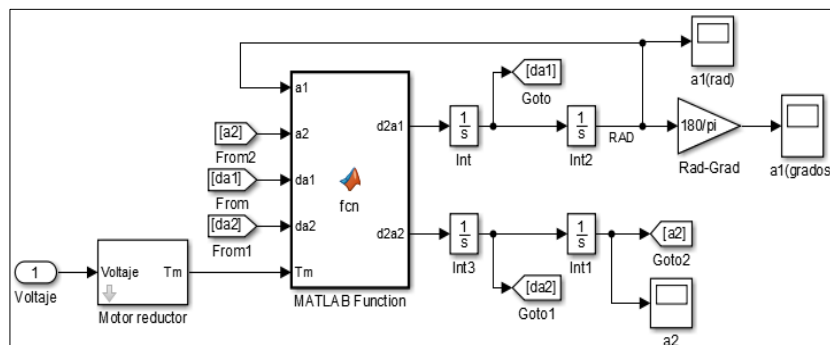
Descripción	Variable	Valor	Unidad de Medida
Masa de la pelota	$m_2$	0.1	$[kg]$
Masa de la rueda	$m_3$	0.25	$[kg]$
Masa del cuerpo	$m_1$	0.7	$[kg]$
Radio de la pelota	$r_e$	0.21	$[m]$
Radio de la rueda	$r_r$	0.052	$[m]$
Inercia de la pelota	$I_e$	0.00294	$[kgm^2]$
Inercia de la rueda	$I_r$	0.000338	$[kgm^2]$
Altura	$b$	0.190	$[m]$
Profundidad	$c$	0.125	$[m]$
Inercia del cuerpo	$I_c$	0.003017	$[kgm^2]$
Aceleración gravitacional	$g$	9.800	$[m/s^2]$

Fuente: Autor

### 3.4 DISEÑO DE CONTROLADORES PID

El ballbot emplea dos funciones de transferencia indicadas en la Figura 3.11, donde el primer bloque consiste en determinar la relación torque y voltaje del motor; en cambio el segundo bloque abarca la relación entre aceleración y torque, este último emplea el algoritmo Lagrangiano indicado en el Anexo 2 junto con los parámetros de la Tabla 3.4. En segunda instancia, se ingresa la salida de la aceleración a dos integrales laplacianos para determinar la posición en radianes o en grados.

Figura 3.11. Función de transferencia total

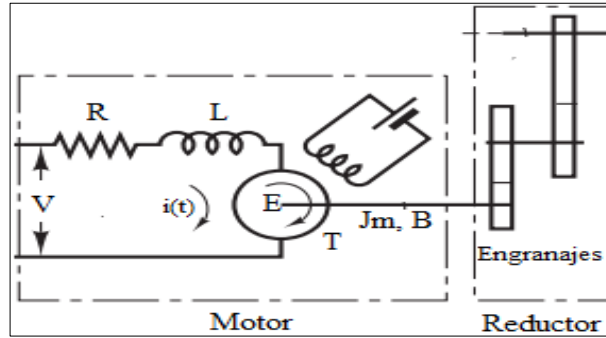


Fuente: Autor

### 3.4.1 Función de transferencia del motor reductor

El motor se compone de una carga eléctrica RLE (resistencia, inductancia, voltaje generado) y carga mecánica desarrollada en el rotor, conectado al eje del motor de corriente continua, tal como se indica en la Figura 3.12 (Corona, Abarca, & Mares, 2019).

Figura 3.12. Circuito eléctrico y mecánico del motor reductor



Fuente: (Corona, Abarca, & Mares, 2019)

A continuación, se utilizan las ecuaciones eléctricas y mecánicas en función del tiempo (Corona, Abarca, & Mares, 2019):

$$\begin{cases} V(t) = Ri(t) + L \frac{di}{dt} + E(t) \\ T_m = J \frac{dw(t)}{dt} + Bw(t) \\ E(t) = k_a w(t) \\ T_m(t) = K_m i(t) \end{cases} \quad \text{Ecuación 3.20}$$

Donde:

- $V(t)$  es el voltaje de armadura medido en [V].
- $i(t)$  representa la corriente en el circuito de armadura medido en [A].
- $T_m$  es el torque del motor expresado en [Nm].
- $J$  representa el momento de inercia cuya unidad de medida esta en [ $kgm^2$ ].
- $k_a$  y  $k_m$  son las constantes de la fuerza electromotriz y del par motor; cuyas unidades de medida se expresan en [Vs/rad] y [Nm/A], respectivamente.

- $w(t)$  es la velocidad angular determinada en [rad/s].
- $B$  es el coeficiente de fricción expresado en [Nms/rad].
- $E(t)$  es el voltaje en el rotor medido en [V].

Luego, se transforman las ecuaciones mediante el Laplaciano (Bejar, 2018):

$$\begin{cases} V(s) = (R + Ls)i(s) + E(s) \\ T_m(s) = (B + Js)w(s) \\ E(s) = k_a w(s) \\ T_m(s) = K_m i(s) \end{cases} \quad \text{Ecuación 3.21}$$

Como siguiente punto, se determina la función de transferencia del torque con relación al voltaje de armadura:

$$\begin{aligned} V(s) &= (R + Ls)i(s) + k_a w(s) \\ V(s) &= \frac{(R + Ls)T_m(s)}{K_m} + \frac{k_a T_m(s)}{(B + Js)} \\ \frac{T_m(s)}{V(s)} &= \frac{K_m(B + Js)}{(R + Ls)(B + Js) + K_m K_a} \end{aligned}$$

Por lo tanto, en la Ecuación 3.22 se obtiene:

$$\frac{T_m(s)}{V(s)} = \frac{K_m(B + Js)}{LJs^2 + (BL + Rj)s + (BR + K_m K_a)} \quad \text{Ecuación 3.22}$$

Las variables eléctricas y mecánicas de la función de transferencia para el motor reductor se calculan de la siguiente manera:

**Paso 1.** La Inercia  $J$  representa la suma de sus inercias del motor  $J_m$  y carga  $J_{carga}$ , donde  $J_m$  es 15 veces menor a  $J_{carga}$  y se calcula por medio de la Ecuación 3.23 (Contaval, 2018).

$$J = J_m + J_{carga} = \frac{16J_{carga}}{15} [kgm^2] \quad \text{Ecuación 3.23}$$

La inercia de carga es igual a la sumatoria de las inercias del cuerpo del robot  $I_c$ , pelota  $I_e$  y rueda virtual  $I_r$ , visto en Ecuación 3.24 (Contaval, 2018).

$$J_{carga} = (I_c + I_e + I_r) [kgm^2] \quad \text{Ecuación 3.24}$$

Reemplazando la Ecuación 3.24 en la 3.23 se obtiene que la inercia  $J$  posee un valor de  $0.006715 [kgm^2]$ .

$$J = \frac{16(I_c + I_e + I_r)}{15} = 0.006715 [kgm^2]$$

**Paso 2.** El torque  $T_m$  y la corriente  $i$  presentan valores de  $0.43 [Nm]$  y  $1.8 [A]$ , respectivamente por lo que la constante del motor  $K_m$  se calcula mediante la Ecuación 3.25 (Philohome, 2020).

$$K_m = \frac{T_m}{i} \left[ \frac{Nm}{A} \right] \quad \text{Ecuación 3.25}$$

Por tanto,  $K_m$  es igual a  $0.24 [Nm/A]$ .

**Paso 3.** Se evalúa el circuito eléctrico de corriente continua RLE de la Figura 3.12, donde la inductancia actúa como cortocircuito, por lo que se asume un valor de  $0 [mH]$ . Además, para efectos de cálculo (criterio de diseño) se anula el voltaje del rotor  $E$ , quedando la Ecuación 3.26.

$$R = \frac{V}{I} \quad \text{Ecuación 3.26}$$

El Ballbot EV3 dispone de un voltaje de alimentación al motor de  $9 [V]$  y una corriente continua  $I = 1.8 [A]$  (Philohome, 2020). De tal manera que la resistencia  $R$  es igual a  $5 [\Omega]$ .

**Paso 4.** La corriente sin carga  $i_{sc}$  y la velocidad del motor  $w$  poseen valores de  $60 [mA]$  y  $175 [rpm]$  ( $18.33 [rad/s]$ ), respectivamente. Además, el voltaje  $E$  se calcula como al producto entre la constante de la fuerza electromotriz  $K_a$  y la velocidad del motor, visto en la Ecuación 3.27 (Philohome, 2020).

$$E = K_a w \quad \text{Ecuación 3.27}$$

La Ecuación 3.27 se reemplaza en el circuito de corriente continua, pero incluyendo el voltaje del rotor  $E$  para calcular  $K_a$  mediante la Ecuación 3.28.



$$K_a = \frac{V - i_{sc}R}{w} \quad \text{Ecuación 3.28}$$

Obteniendo un valor de 0.475 [Vs/rad] de la constante  $K_a$ .

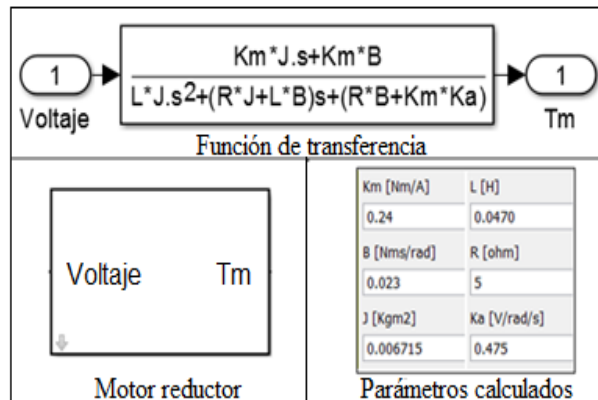
**Paso 5.** La fricción  $B$  se calcula como la razón entre el torque  $T_m$  y la velocidad del motor tal como se indica en la Ecuación 3.29 (Philohome, 2020).

$$B = \frac{T_m}{w} \left[ \frac{Nms}{rad} \right] \quad \text{Ecuación 3.29}$$

Por tanto,  $B$  es igual a 0.023 [Nms/rad].

Las ecuaciones calculadas son reemplazadas en la Ecuación 3.22 e ingresadas a la función de transferencia del motor, tal como se muestra en la Figura 3.13.

Figura 3.13. Función de transferencia del motor reductor



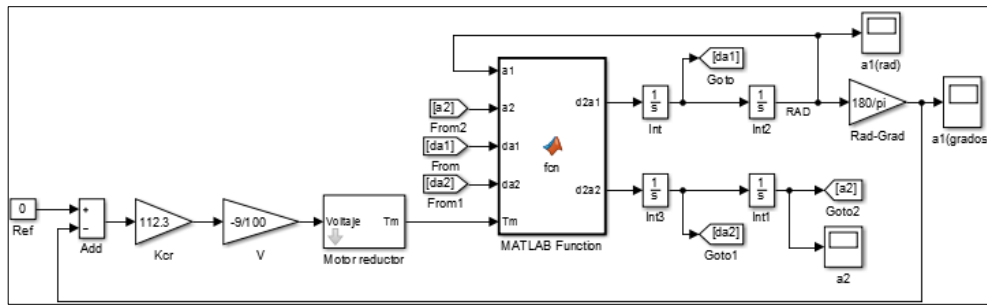
Fuente: Autor

Cabe mencionar que la función de transferencia contiene el valor de la inductancia del circuito eléctrico, por lo que se toma como factor de diseño a uno similar de 47 [mH] que proporciona el motor NXT de Lego Mindstorms, puesto que no se dispone de este valor en el EV3 (Golnaraghi, 2012).

### 3.4.2 Sintonización PID mediante Ziegler Nichols

En la Figura 3.14 se presenta el diseño PID de Ziegler Nichols basado en el método de la ganancia crítica o lazo cerrado de las funciones de transferencia del Ballbot EV3 a través de la herramienta Simulink de Matlab. Como se indicó en el Apartado 2.5.2, se emplea una constante proporcional desde 1 unidad y se incrementa hasta que el sistema presente oscilaciones sostenidas.

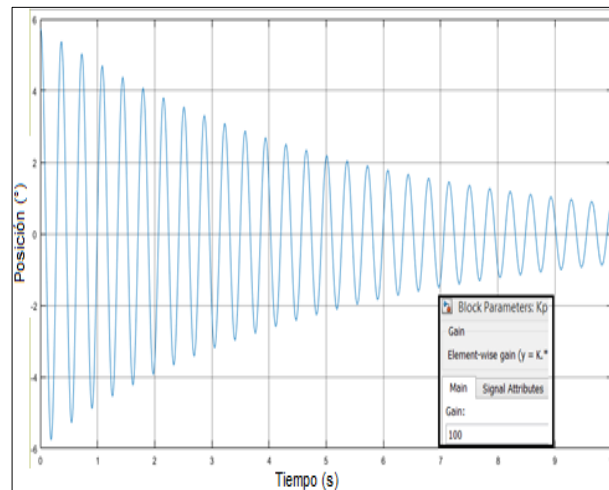
Figura 3.14. Sistema del robot en lazo cerrado



Fuente: Autor

Como es un proceso de simulación, se realizan tres pruebas con ganancias entre 100 a 113 unidades. En la Figura 3.15, se observa que las oscilaciones disminuyen al paso del tiempo para una constante proporcional de 100 unidades.

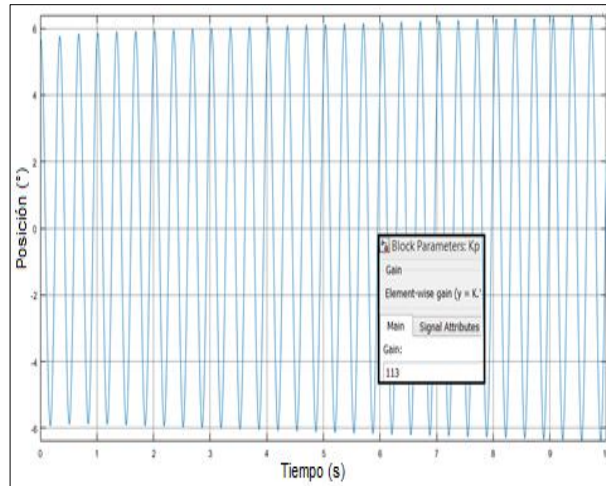
Figura 3.15. Respuesta del sistema en lazo cerrado con  $K_p = 100$



Fuente: Autor

En cambio, para una constante proporcional de 113 unidades se observa en la Figura 3.16 que las amplitudes de las oscilaciones se incrementan en el tiempo y no se estabilizan.

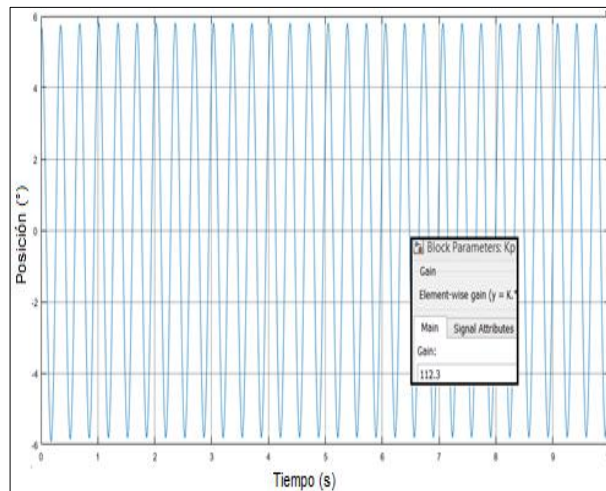
Figura 3.16. Respuesta del sistema en lazo cerrado con  $K_p = 113$



Fuente: Autor

Por último, se utiliza una constante proporcional de 112.3 unidades y se observa en la Figura 3.17 que las amplitudes de las oscilaciones se mantienen constantes, por lo que se determina que la ganancia crítica  $K_{cr}$  es igual a 112.3 para un periodo crítico  $P_{cr}$  de 335 [ms], estos valores se reemplazan en la Tabla 2.2.

Figura 3.17. Respuesta del sistema en lazo cerrado con  $K_p = 112.3$



Fuente: Autor

Se realiza una sustitución de las variables  $K_p$ ,  $T_i$  y  $T_d$  de la Ecuación 2.5 por las constantes  $P$ ,  $I$  y  $D$  de la Ecuación 3.30, obteniendo los siguientes valores:

$$G_c(s) = K_p \left[ 1 + \frac{1}{T_i s} + T_d s \right] = P + \frac{I}{s} + Ds$$

Donde:

$$P = K_p = 0.6K_{cr} = 0.6 (112.3) = 67.38$$

$$I = \frac{K_p}{T_i} = \frac{67.38}{2P_{cr}} = \frac{67.38}{2(0.335)} = 402.269$$

$$D = K_p T_d = 0.125(67.38)(0.335) = 2.82$$

### 3.4.3 Diseño del controlador PID difuso

El clásico controlador PID en el tiempo continuo viene dado por la Ecuación 3.30 en el dominio de la frecuencia mediante la transformada de Laplace (Hernández, 2010).

$$U(s) = \left( P + \frac{I}{s} + Ds \right) E(s) \quad \text{Ecuación 3.30}$$

Luego, se realiza una conversión del tiempo continuo a discreto y una sustitución de variables por medio de la Ecuación 3.31 (Carvajal, Chen, & Ogmen, 2000).

$$s = \frac{2}{T} \left( \frac{1 - z^{-1}}{1 + z^{-1}} \right), \quad E'(z) = sE(s) \quad \text{Ecuación 3.31}$$

Donde el tiempo de muestreo  $T$  debe ser mayor a cero; por consiguiente, se obtiene el controlador PID en tiempo discreto (Carvajal, Chen, & Ogmen, 2000).

$$U(z) = \left( P + \frac{IT}{2} \left( \frac{1 + z^{-1}}{1 - z^{-1}} \right) \right) E(z) + DE'(z) \quad \text{Ecuación 3.32}$$

Se elimina el operador  $(1 - z^{-1})$  multiplicándolo a ambas partes de la Ecuación 3.33, luego se calcula la transformada inversa, obteniendo la siguiente ecuación:

$$U = I_1 e(kT) + P_1 (e(kT) - e(kT - T)) + De' \quad \text{Ecuación 3.33}$$

Donde:

$$U = U(kT) - U(kT - T)$$

$$k = 0, 1, 2, 3, \dots$$

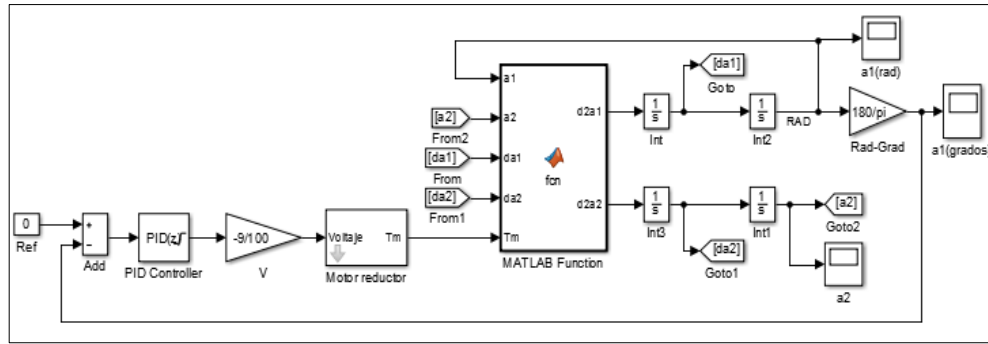
$$P_1 = P - \frac{I_1 T}{2}$$

$$I_1 = IT$$

$$e' = e'(kT) - e'(kT - T)$$

Pues bien, con la ayuda del PID Tuner del Simulink de Matlab, se obtienen las constantes proporcional, integral y derivativa. Además, se coloca una referencia o set point de cero; tal como se muestra en la Figura 3.18.

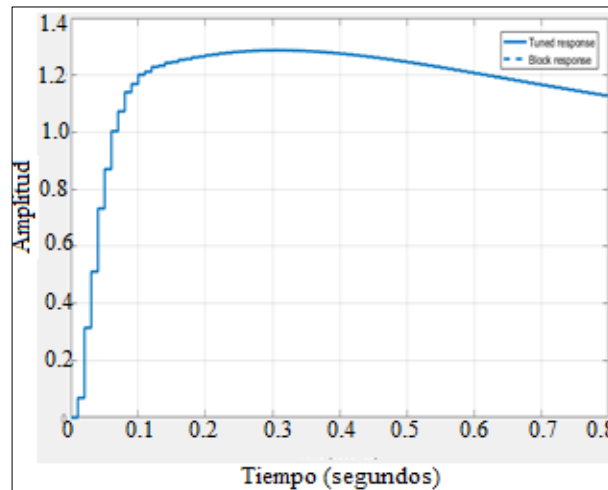
Figura 3.18. Diagrama de bloques del sistema con PID Tuner



Fuente: Autor

Se escoge un PID discreto cuyo tiempo de muestreo es de 10 milisegundos y presenta los siguientes valores negativos: proporcional  $P = 55.5$ , integral  $I = 97.9$  y derivativo  $D = 7.3$ , indicado en la Figura 3.19.

Figura 3.19. Respuesta del sistema con el PID Tuner



Fuente: Autor

La tarea del PID difuso consiste en determinar un conjunto idóneo de comandos para que el robot alcance el estado deseado sin problemas y desviaciones mínimas. De las Ecuaciones 3.34 a la 3.36 se expresa el error de seguimiento  $e_I$ , cambio en el error  $e_P$  y cambio en la tasa de error  $e_D$ , respectivamente. Estos términos representan las señales de entrada, en cambio la salida de control incremental representa  $U(kT) - U(kT - T)$  indicado en la Ecuación 3.37 (Carvajal, Chen, & Ogmen, 2000).

$$e_I = I_1 e(kT) \quad \text{Ecuación 3.34}$$

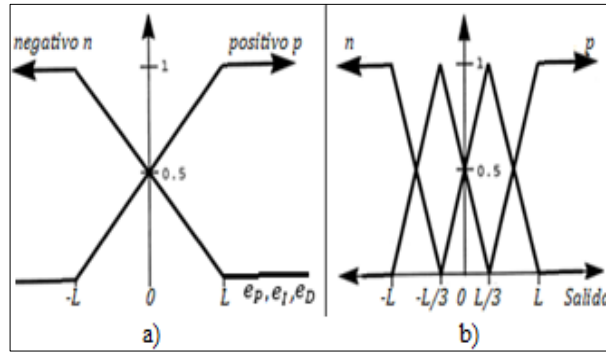
$$e_P = P_1(e(kT) - e(kT - T)) \quad \text{Ecuación 3.35}$$

$$e_D = D(e'(kT) - e'(kT - T)) \quad \text{Ecuación 3.36}$$

$$\Delta u = e_I + e_P + e_D = U(kT) - U(kT - T) \quad \text{Ecuación 3.37}$$

Para el ballbot, la función de pertenencia para las entradas se aprecia en la Figura 3.20.a, y las funciones de membresía de la salida en la Figura 3.20.b; ya que poseen varios resultados. Por tanto, se definen ocho reglas acordes al parámetro de umbral  $L$ , este término indica el valor mínimo y máximo para el proceso de fuzzificación (Carvajal, Chen, & Ogmen, 2000).

Figura 3.20. Funciones a) pertenencia (entradas) y b) membresía (salida)



Fuente: (Carvajal, Chen, & Ogmen, 2000)

A continuación, se presenta las 8 reglas de composición de enlace (Carvajal, Chen, & Ogmen, 2000):

- **R1.** Sí  $e_P \in e_P \cdot n$  &  $e_I \in e_I \cdot n$  &  $e_D \in e_D \cdot n$ , entonces  $\Delta u \in \Delta u \cdot nl$ .
- **R2.** Sí  $e_P \in e_P \cdot n$  &  $e_I \in e_I \cdot n$  &  $e_D \in e_D \cdot p$ , entonces  $\Delta u \in \Delta u \cdot n$ .
- **R3.** Sí  $e_P \in e_P \cdot p$  &  $e_I \in e_I \cdot n$  &  $e_D \in e_D \cdot n$ , entonces  $\Delta u \in \Delta u \cdot n$ .
- **R4.** Sí  $e_P \in e_P \cdot p$  &  $e_I \in e_I \cdot n$  &  $e_D \in e_D \cdot p$ , entonces  $\Delta u \in \Delta u \cdot p$ .
- **R5.** Sí  $e_P \in e_P \cdot n$  &  $e_I \in e_I \cdot p$  &  $e_D \in e_D \cdot n$ , entonces  $\Delta u \in \Delta u \cdot n$ .
- **R6.** Sí  $e_P \in e_P \cdot n$  &  $e_I \in e_I \cdot p$  &  $e_D \in e_D \cdot p$ , entonces  $\Delta u \in \Delta u \cdot p$ .
- **R7.** Sí  $e_P \in e_P \cdot p$  &  $e_I \in e_I \cdot p$  &  $e_D \in e_D \cdot n$ , entonces  $\Delta u \in \Delta u \cdot p$ .
- **R8.** Sí  $e_P \in e_P \cdot p$  &  $e_I \in e_I \cdot p$  &  $e_D \in e_D \cdot p$ , entonces  $\Delta u \in \Delta u \cdot pl$ .

Donde ( $\in$ ) representa la pertenencia, (&) significa la operación lógica “y”, ( $\cdot$ ) es un símbolo que representa la combinación y el término ( $l$ ) indica una magnitud grande

ya sea positiva o negativa. Si en cada regla se cumplen dichas condiciones, implica que la salida pertenece a ese grupo mencionado. Luego de este proceso, se aplica la defuzzificación mediante el método Sugeno o centro de gravedad, tal como se indica en las Ecuaciones 3.38 y 3.39, donde  $u(Ri)$  representa el valor de pertenencia (Carvajal, Chen, & Ogmen, 2000).

$$\Delta u = \frac{\sum_{i=1..8}\{membresía(entrada_i) * salida_i\}}{\sum_{i=1..8}\{membresía(entrada_i)\}} \quad \text{Ecuación 3.38}$$

$$\Delta u = \frac{u(R1) * nl + u(R2) * l + \dots + u(R7) * p + u(R8) * pl}{u(R1) + u(R2) + u(R3) + \dots + u(R6) + u(R7) + u(R8)} \quad \text{Ecuación 3.39}$$

Partiendo de la Figura 3.20.a, se tiene como ejemplo el cálculo de la pendiente  $e_p \cdot p$ , visto en la Ecuación 3.40 (Carvajal, Chen, & Ogmen, 2000).

$$e_p \cdot p = \frac{e_p}{L - (-L)} + 0.5 = \frac{e_p + L}{2L} \quad \text{Ecuación 3.40}$$

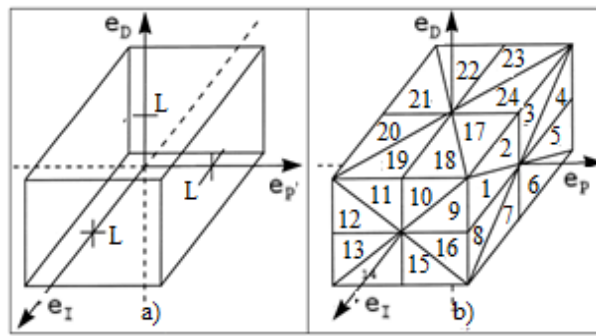
Posteriormente, se definen las funciones de pertenencia individuales tal como se muestran en las Ecuaciones 3.41 y 3.42 (Carvajal, Chen, & Ogmen, 2000).

$$e_p \cdot n = \frac{-e_p + L}{2L}; e_l \cdot n = \frac{-e_l + L}{2L}; e_D \cdot n = \frac{-e_D + L}{2L} \quad \text{Ecuación 3.41}$$

$$e_p \cdot p = \frac{e_p + L}{2L}; e_l \cdot p = \frac{e_l + L}{2L}; e_D \cdot p = \frac{e_D + L}{2L} \quad \text{Ecuación 3.42}$$

Debido a que el controlador PID posee tres componentes diferentes a la entrada, existen 8 combinaciones que dieron origen a las 8 reglas de composición de enlace, por tanto, pueden ser representadas en un cubo con la limitación del umbral, tal como se indica la Figura 3.21.a. Las reglas de la defuzzificación se construyen dividiendo al cubo en 48 partes, visto en la Figura 3.21.b (Carvajal, Chen, & Ogmen, 2000).

Figura 3.21. Función de pertenencia difusa

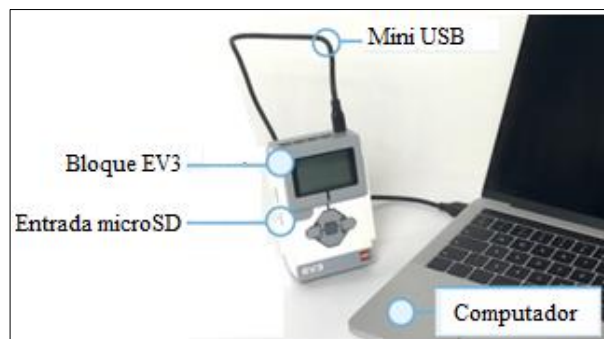


Fuente: (Carvajal, Chen, & Ogmen, 2000)

### 3.5 SOFTWARE EV3

El software empleado para el control del ballbot EV3 se realiza a través del micro Python. Para ello, se instala la imagen de EV3 MicroPython en una tarjeta microSD para ser ingresada el bloque EV3 y luego conectarse al computador, tal como se indica en la Figura 3.22 (LEGO, 2019).

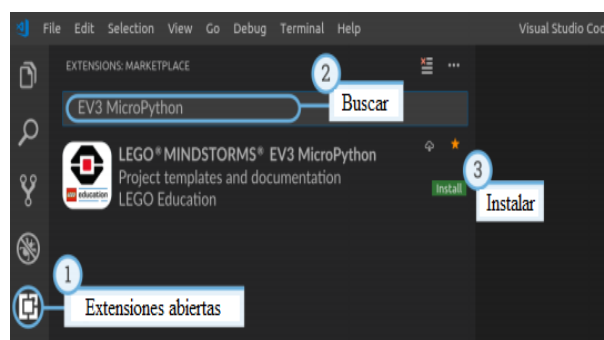
Figura 3.22. Conexión para la configuración MicroPython



Fuente: (LEGO, 2019)

Por consiguiente, se instala el EV3 MicroPython en el Visual Studio Code, visto en la Figura 3.23.

Figura 3.23. MicroPython en Visual Studio Code



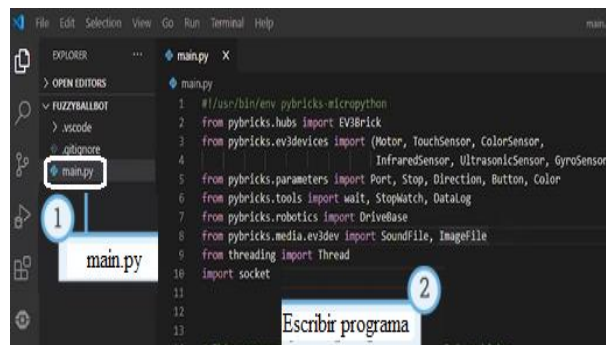
Fuente: (LEGO, 2019)

A continuación, se describe una serie de pasos para elaborar el diseño del controlador PID difuso en el lenguaje MicroPython:



**Paso 1.** En la creación de un nuevo proyecto en el Visual Studio Code, se incluye por defecto el archivo main.py; dentro de este se agrega el software para el control del Ballbot. La librería Pybricks permite que el usuario posea acceso a todas las funcionalidades que tiene el bloque EV3, visto en la Figura 3.24 (LEGO, 2019). Adicional se utiliza la librería “threading” de python e importación de un socket para el control remoto.

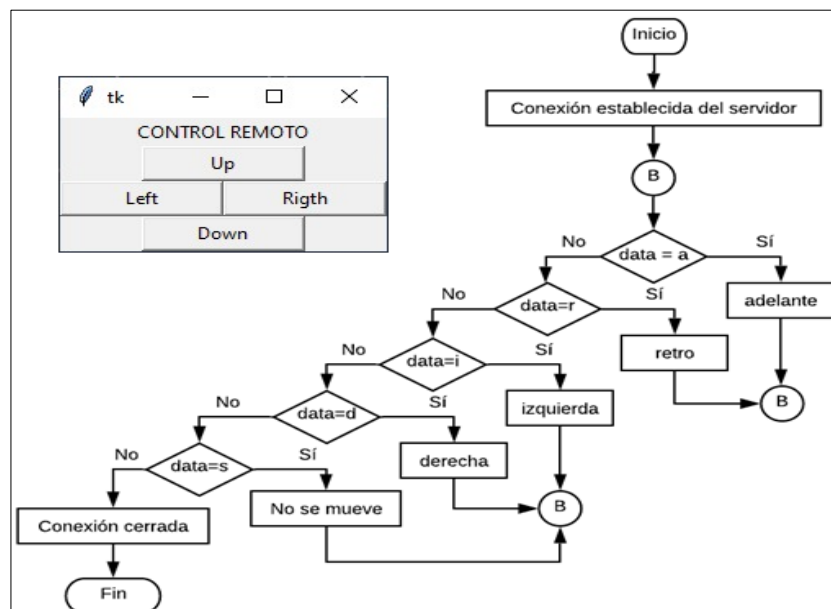
Figura 3.24. Encabezado principal del programa EV3brick



Fuente: Anexo 3

**Paso 2.** Se define el puerto del servidor con comunicación TCP/IP, sockets y los movimientos del robot (adelante “a=UP”, hacia atrás o retro “r=Down”, izquierda “i=Left” y derecha “d=Right”), donde la variable denominada “data” detecta y realiza el movimiento siguiente la lógica indicada en la Figura 3.25. Además, los movimientos son controlados de forma inalámbrica a través de la interfaz gráfica Tkinter (Anexo 4) desde el computador.

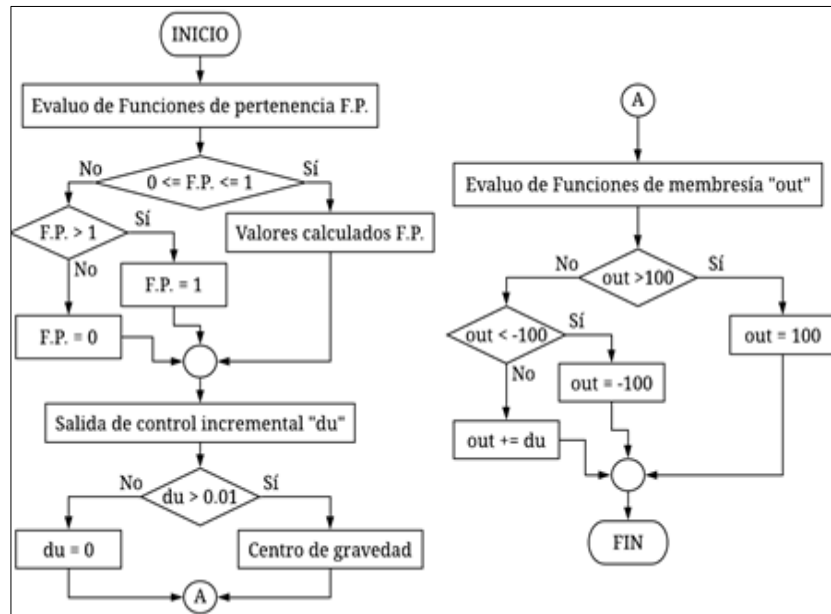
Figura 3.25. Movimientos del Ballbot EV3



Fuente: Anexos 3 y 4

**Paso 3.** Se calculan las funciones de pertenencia proporcional, integral y derivativo, por medio de la cual se determina el valor de salida incremental. Luego, se compara con las 8 reglas de composición de enlace y se determina el centro de gravedad para la defuzzificación, donde se obtiene la salida o función de membresía del proceso, visto en la Figura 3.26.

Figura 3.26. Cálculo de la salida del control PID difuso



Fuente: Anexo 4

**Paso 4.** Se define una clase PID difusa que contienen 2 instancias pid1 y pid2, cuyo propósito es el control de los 2 ángulos y poder mantener el equilibrio. Además, se inicializa las variables de los ángulos a cero grados con un tiempo de muestro de 10 ms.

**Paso 5.** El funcionamiento del robot consiste en la detección de posición de los sensores de giro, por consiguiente, se determinan los errores y se calcula la salida, tal como se indicó en la Figura 3.26. En caso de existir una condición de desplazamiento (Figura 3.25), se mueve la referencia hacia la dirección en que se necesita realizar el movimiento, además se agrega un limitador del voltaje de caída inicial del motor para disminuir el efecto de la inercia que el robot tiene al comenzar un desplazamiento entre 35 a 40 unidades.

## CAPÍTULO 4

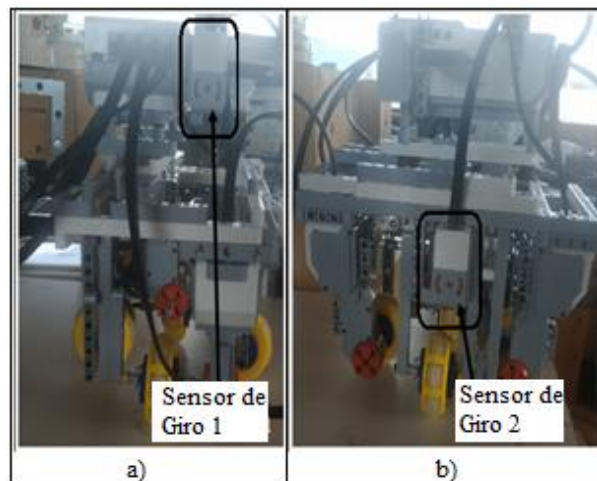
### PRUEBAS Y RESULTADOS

En este esquema capitular se realiza las pruebas de la sintonización del PID de Ziegler Nichols y PID difuso en el Ballbot EV3.

#### 4.1 PRUEBAS DE EQUILIBRIO CON EL CONTROLADOR PID ZIEGLER NICHOLS

El Ballbot EV3 dispone de 2 sensores de giro ubicados en la Figura 4.1.a) y 4.1.b) para las mediciones del ángulo 1 y 2, respectivamente.

Figura 4.1. Posición de los sensores de giro



Fuente: Autor

Además, se escoge las variables PID del Apartado 3.4.2, tal que:  $P = 67.4$ ,  $I = 402.3$  y  $D = 2.8$ ; estas se ingresan al controlador PID en el software MicroPython y determina datos promedios referentes a las mediciones de los ángulos 1 y 2 con sus respectivos voltajes entre -9 a 9 [V]. Cabe mencionar que el bloque EV3 lee esos valores de voltaje como -100 a +100 [unidades], es decir que -9 [V] representa -100 [unidades] y 9 [V] es igual a +100 [unidades].

Tabla 4.1. Ángulo de giro y voltaje con PID Ziegler Nichols

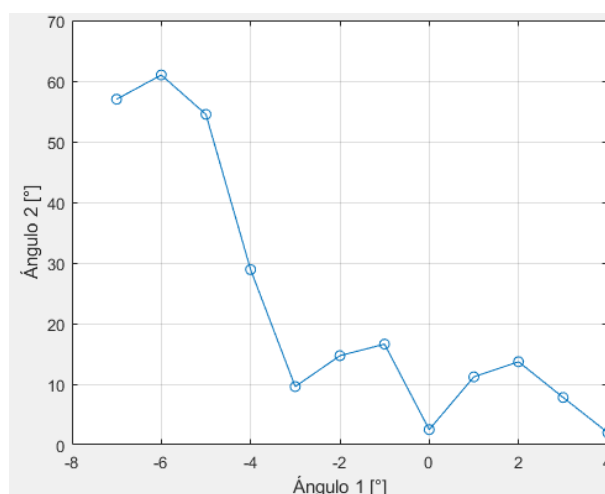
Conteo datos Ángulo 1	Ángulo 1 [°]	Salida Voltaje 1 [unidades]	Ángulo 2 [°]	Salida Voltaje 2 [unidades]
2	-7.0	-100.0	57.0	-100.0
4	-6.0	-100.0	61.0	-100.0
6	-5.0	-100.0	54.5	-100.0
15	-4.0	-100.0	28.9	-46.7
26	-3.0	-100.0	9.6	-3.8
30	-2.0	-100.0	14.7	-13.3
36	-1.0	-99.7	16.6	-22.2
81	0.0	0.0	2.5	-13.6
39	1.0	100.0	11.2	-56.4
33	2.0	100.0	13.7	-84.8
21	3.0	100.0	7.8	-75.5
3	4.0	100.0	2.0	-100.0

Fuente: Autor

Para que el Ballbot se encuentre en equilibrio, los ángulos de giro deben ser cero grados; por lo que se realiza el siguiente análisis:

- En la Figura 4.2 se expone una comparación de los 2 giros, donde se observa que los ángulos de giro poseen una alta desviación y se alejan de los 0°.

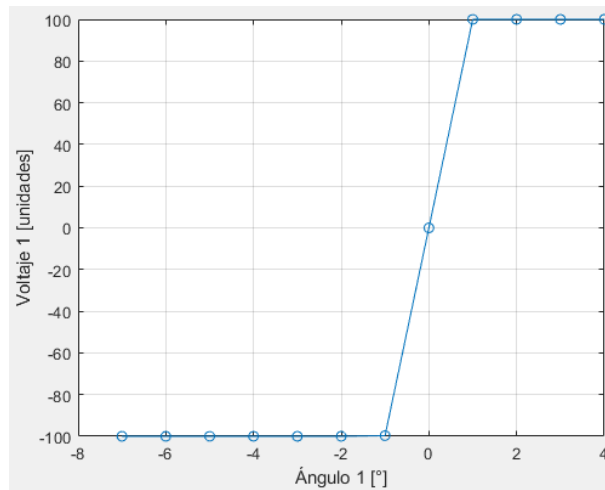
Figura 4.2. Comparación de los ángulos de giro con PID Ziegler Nichols



Fuente: Autor

- En la Figura 4.3 se observa que la salida de voltaje utilizando el sensor de giro 1 es de -100 [unidades] por cada grado (negativo); en cambio, si es +100 [unidades] por lo que el control PID aplicado no reacciona ante los movimientos del robot.

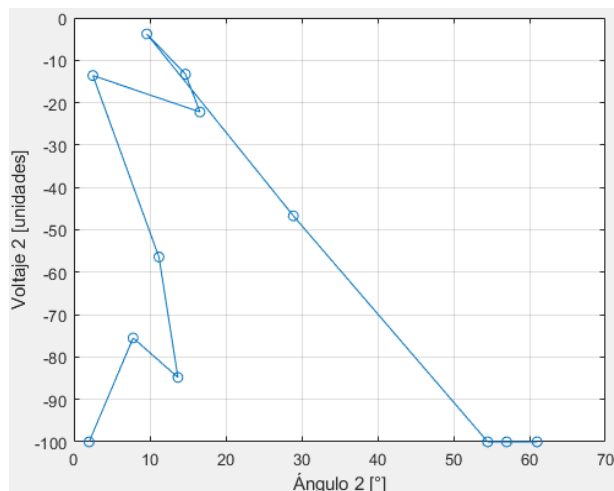
Figura 4.3. Giro vs Voltaje 1 con Ziegler Nichols



Fuente: Autor

- En la Figura 4.4 se observa cómo cambia de forma radical la salida de voltaje ante el giro del ángulo 2; por lo que tiene mayor probabilidad de que el Ballbot EV3 no tenga equilibrio.

Figura 4.4. Giro vs Voltaje 2 con Ziegler Nichols



Fuente: Autor

## 4.2 PRUEBAS DE EQUILIBRIO CON EL CONTROLADOR PID DIFUSO

### 4.2.1 Prueba inicial

Se escoge las variables PID del Apartado 3.4.3, tal que:  $P = 55.5$ ,  $I = 97.9$  y  $D = 7.3$ ; estas se ingresan al controlador PID junto con las reglas difusas en el software MicroPython y determina datos promedios referentes a las mediciones de los ángulos 1 y 2 con sus respectivos voltajes, tal como se observa en la Tabla 4.2.

Tabla 4.2. Mediciones del giro y voltaje con el PID difuso inicial

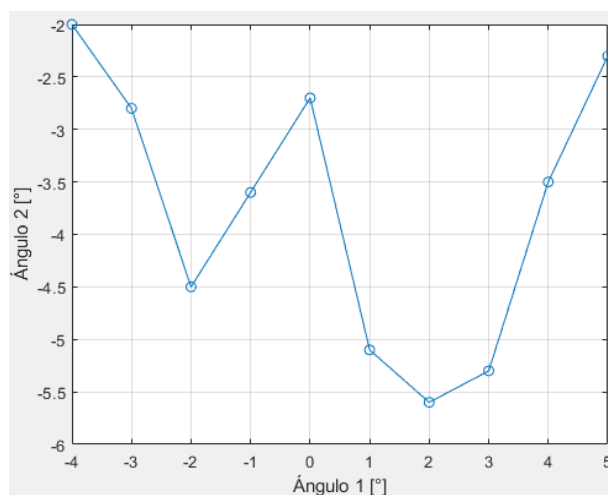
Conteo datos Ángulo 1	Ángulo 1 [°]	Salida Voltaje 1 [unidades]	Ángulo 2 [°]	Salida Voltaje 2 [unidades]
1	-4.0	-100.0	-2.0	100.0
6	-3.0	-100.0	-2.8	77.6
26	-2.0	-100.0	-4.5	59.6
53	-1.0	-96.7	-3.6	26.6
56	0.0	0,0	-2.7	7.7
44	1.0	69.2	-5.1	33.3
41	2.0	100.0	-5.6	44.9
29	3.0	100.0	-5.3	37.4
14	4.0	100.0	-3.5	11.6
3	5.0	100.0	-2.3	-33.3

Fuente: Autor

A continuación, se realiza el siguiente análisis:

- En la Figura 4.5 se observa como una diferencia de ángulos de  $9^\circ$  y  $-3.6^\circ$  con respecto a los ángulos 1 y 2, por lo que el robot trata de mantener el equilibrio ya que sus ángulos son cercanos a  $0^\circ$ .

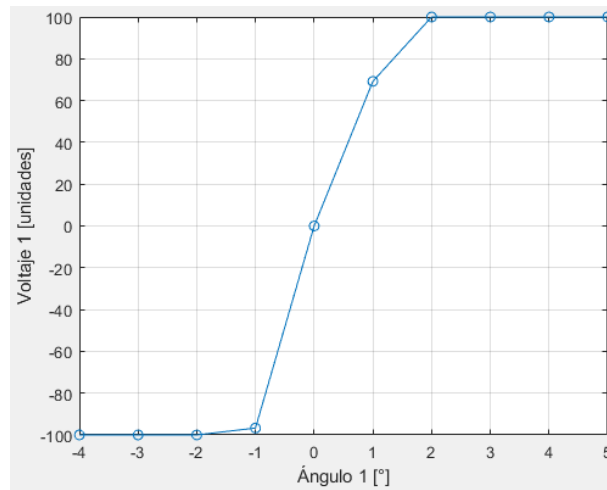
Figura 4.5. Comparación del ángulo de giro con PID difuso inicial



Fuente: Autor

- Con respecto a la variación del ángulo 1 medido por el sensor, se tiene una variación entre  $-1^\circ$  a  $1^\circ$  cuyos valores de voltaje entre  $-96.7$  a  $+69.7$  [unidades], tal como se indica en la Figura 4.6. Sin embargo, fuera de ese rango el controlador no responde adecuadamente y pierde el equilibrio debido al aumento de la constante integral del PID.

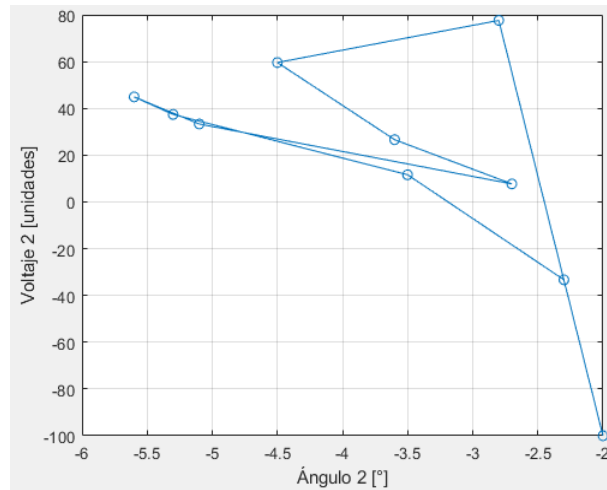
Figura 4.6. Giro vs Voltaje 1 con PID difuso inicial



Fuente: Autor

- Similar al caso de Ziegler Nichols, no posee un buen control sobre el ángulo 2 de giro, tal como se indica en la Figura 4.7.

Figura 4.7. Giro vs Voltaje 2 con PID difuso inicial



Fuente: Autor

#### 4.2.2 Calibración final

Es necesario realizar una calibración adicional para que el robot pueda mantener el equilibrio, por lo que se realiza un ajuste en las constantes proporcionales  $P = 25.0$ , integrales  $I = 105.0$  y derivativas  $D = 0.3$ . Además, se incluyen las reglas difusas y determina datos promedios referentes a las mediciones de los ángulos de giro 1 y 2 con sus respectivos voltajes entre  $-9$  [V] o  $-100$  [unidades] a  $9$  [V] o  $100$  [unidades], visto en la Tabla 4.3.

Tabla 4.3. Mediciones del giro y voltaje con el PID difuso final

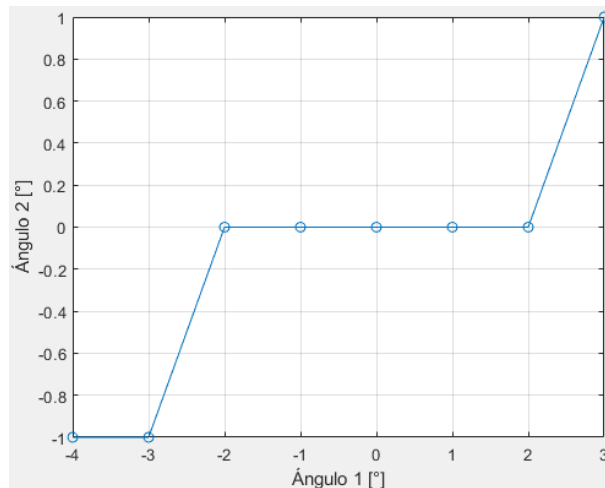
Conteo datos	Ángulo 1	Ángulo 1 [°]	Salida Voltaje 1 [unidades]	Ángulo 2 [°]	Salida Voltaje 2 [unidades]
3		-4.0	-100.0	-1.0	-6.2
4		-3.0	-61.7	-1.0	-8.0
1		-2.0	-25.3	0.0	0.0
196		-1.0	-42.7	0.0	-3.3
901		0.0	0.0	0.0	-2.2
394		1.0	44.1	0.0	-2.8
10		2.0	81.6	0.0	0.5
3		3.0	100.0	1.0	-44.1

Fuente: Autor

A continuación, se realiza el siguiente análisis:

- En la Figura 4.8 se observa como la diferencia de ángulos ha disminuido con respecto al PID difuso inicial; por lo que sus valores presentan una diferencia de 7° y 2° en los ángulos 1 y 2, respectivamente.

Figura 4.8. Comparación de giro con el PID difuso final

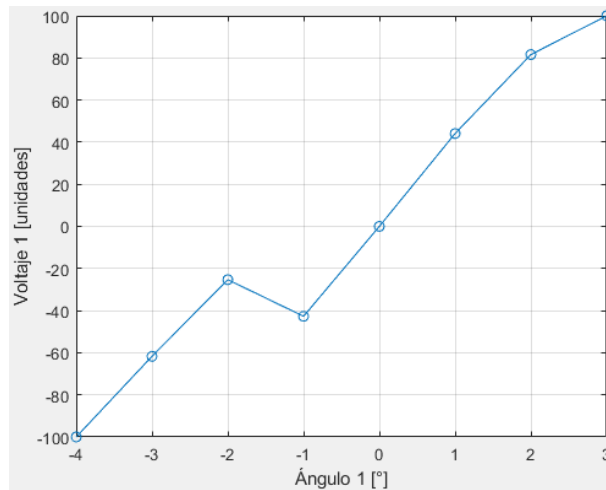


Fuente: Autor

- Con respecto a la variación del ángulo 1 medido por el sensor, se tiene una variación entre -3° a 2° cuyos valores de voltaje están entre -61.7 a +81.6 [unidades], tal como se indica en la Figura 4.9. Por tanto, este controlador PID responde adecuadamente según los cambios medidos por el primer sensor ya que por cada posición se aplica un voltaje que está en el rango entre -100 a 100 [unidades].



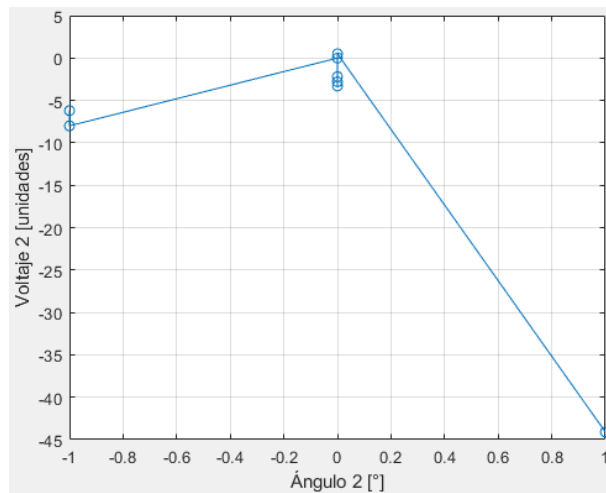
Figura 4.9. Giro vs Voltaje 1 con PID difuso final



Fuente: Autor

- Con respecto a la variación del ángulo 2 medido por el sensor, se tiene una variación entre  $-1^\circ$  a  $+1^\circ$  cuyos valores de voltaje entre  $-44.5$  a  $+0.5$  [unidades], tal como se indica en la Figura 4.10. Por tanto, este controlador PID responde adecuadamente según los cambios medidos por el segundo sensor ya que por cada posición se aplica un voltaje que está en el rango entre  $-100$  a  $100$  [unidades].

Figura 4.10. Giro vs Voltaje 2 con PID difuso final



Fuente: Autor

### 4.2.3 Comparación entre los 2 PID difusos

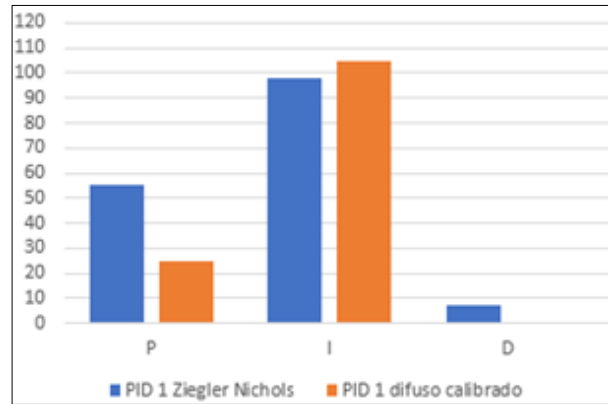
Se observa que las constantes del controlador PID calibradas son menores a lo diseñado del PID Tuner de Simulink; donde la constante proporcional tuvo una variación de 10.5 unidades, la constante integral en 57.9 unidades y la constante derivativa se redujo en 7.2 unidades, tal como se indica en la Tabla 4.4 y Figura 4.11.

Tabla 4.4. Comparación entre constantes PID

Controlador PID	P	I	D
PID 1 inicial	55.5	97.9	7.3
PID 1 final	25.0	105.0	0.3

Fuente: Autor

Figura 4.11. Comparación de constantes del controlador PID 1



Fuente: Autor

En la Tabla 4.5 se observa el intervalo entre los cambios de giro con el uso del controlador PID definido por el Tuner de Simulink y el calibrado, donde se ha reducido en 2° para el giro 1 y en 1.6° para el giro 2.

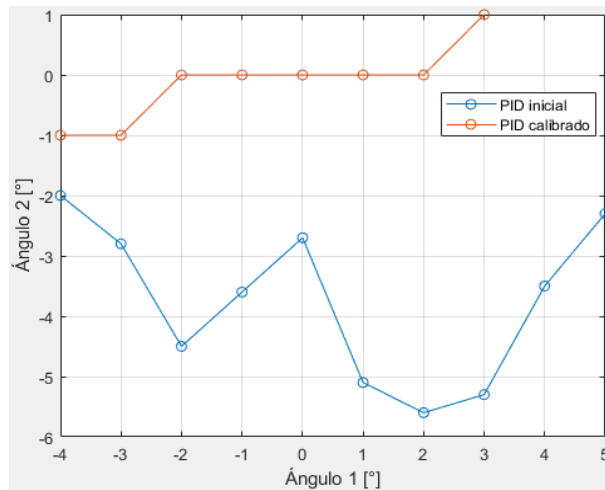
Tabla 4.5. Comparación entre ángulos de giro con PID difuso

Descripción	PID inicial		PID final	
	Ángulo 1 [°]	Ángulo 2 [°]	Ángulo 1 [°]	Ángulo 2 [°]
Medida mínima	-4.0	-5.6	-4.0	-1.0
Medida máxima	5.0	-2.0	3.0	1.0
Variación	9.0	3.6	7.0	2.0

Fuente: Autor

En la Figura 4.12 se observa como el controlador PID calibrado disminuye en los intervalos de los ángulos de giro, haciendo que el Ballbot posea una amplia estabilidad.

Figura 4.12. Comparación de los ángulos de giro con el PID difuso



Fuente: Autor

### 4.3 PRUEBAS Y RESULTADOS ENTRE LOS 2 CONTROLADORES PID

#### 4.3.1 Cambio de variables PID

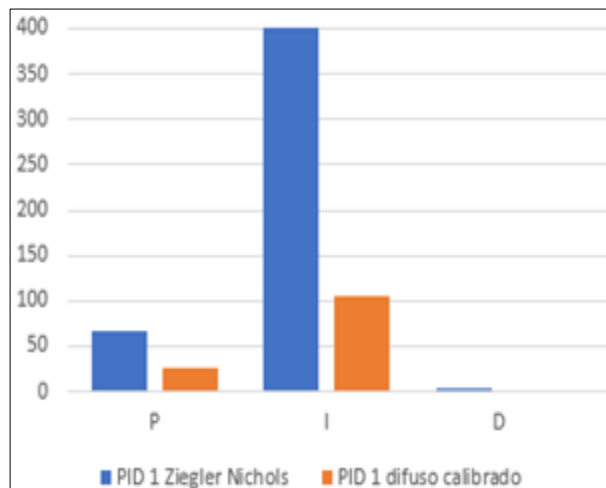
Se observa que las constantes del controlador PID difuso es menor al PID obtenido con el método de Ziegler Nichols; donde la constante proporcional tuvo una variación de 42.4 unidades, la constante integral en 297.3 unidades y la constante derivativa se redujo en 2.5 unidades, tal como se indica en la Tabla 4.6 y Figura 4.13.

Tabla 4.6. Constantes PID Ziegler Nichols vs PID difuso

Controlador PID	P	I	D
PID 1 Ziegler Nichols	67.4	402.3	2.8
PID difuso1	25.0	105.0	0.3

Fuente: Autor

Figura 4.13. Comparación de constantes PID Ziegler Nichols vs PID difuso



Fuente: Autor

### 4.3.2 Ángulos de giro

En la Tabla 4.7 se observa el intervalo entre los cambios de giro con el uso del controlador PID de Ziegler Nichols y PID difuso, donde el ángulo de giro 1 del PID difuso posee una reducción de  $4^\circ$  en el movimiento del robot y  $57^\circ$  con respecto al ángulo de giro 2.

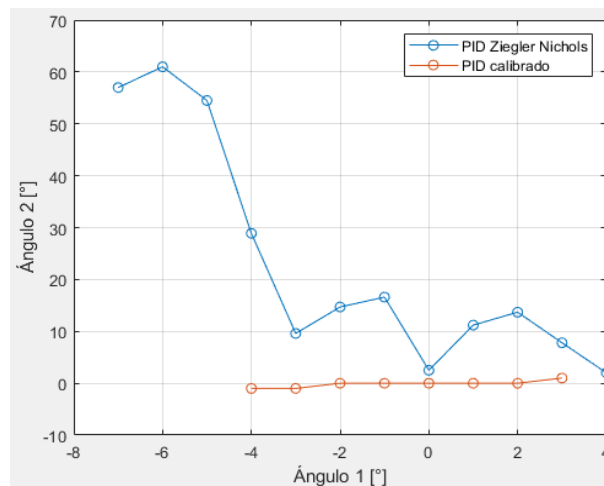
Tabla 4.7. Ángulos de giro con PID Ziegler Nichols vs PID difuso

Descripción	PID Ziegler Nichols		PID Fuzzy	
	Ángulo 1 [ $^\circ$ ]	Ángulo 2 [ $^\circ$ ]	Ángulo 1 [ $^\circ$ ]	Ángulo 2 [ $^\circ$ ]
Medida mínima	-7.0	2.0	-4.0	-1.0
Medida máxima	4.0	61.0	3.0	1.0
Variación	11.0	59.0	7.0	2.0

Fuente: Autor

En la Figura 4.14 se observa como el controlador PID Fuzzy presenta un mejor equilibrio que PID de Ziegler Nichols, ya que los movimientos de giro del robot son relativamente bajos.

Figura 4.14. Ángulos de giro con PID Ziegler Nichols y PID difuso

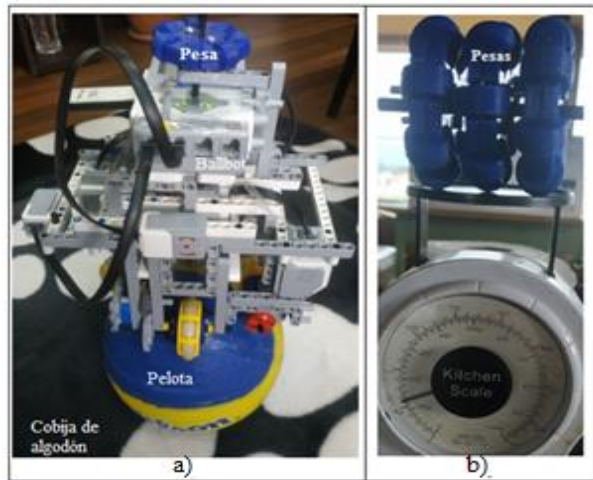


Fuente: Autor

### 4.3.3 Pruebas con diferentes niveles de peso

Se ha realizado varias pruebas experimentales de equilibrio y desplazamiento del Ballbot EV3 sobre un plano horizontal e inclinado, pero utilizando como suelo una cobija de algodón sobre una tabla de medidas 80x70 [cm] (largo x ancho), tal como se indica en la Figura 4.15.a. Las pruebas bajo el plano horizontal fueron utilizadas con el controlador PID de Ziegler Nichols y PID difuso. En cambio, sobre el plano inclinado sólo se optó por el uso del controlador PID difuso con/sin pesos (Figura 4.20.b) ya que el robot mantiene un buen equilibrio con dicha condición.

Figura 4.15. Pruebas de funcionamiento del Ballbot EV3



Fuente: Anexo 6

En la Tabla 4.8 se expone los tiempos de duración ante los controladores PID de Ziegler Nichols y difuso. Cuando se utiliza el PID Ziegler Nichols, posee un equilibrio en un tiempo aproximado de 2 segundos, luego pierde el equilibrio y se cae el robot; en cambio, cuando se utiliza el PID difuso mantiene el equilibrio hasta un peso adicional de 200 [gr].

Tabla 4.8. Equilibrio del Ballbot EV3

Condición PID	Pesas [gr]	Plano	Tiempo (s)			
			Prueba 1	Prueba 2	Prueba 3	Promedio
Ziegler Nichols	0	Horizontal	2	2	2	2
Fuzzy	0	Horizontal	> 20	> 20	> 20	> 20
Fuzzy	0	Inclinado (3°)	> 20	> 21	> 20	> 20
Fuzzy	45	Inclinado (3°)	> 20	> 20	> 20	> 20
Fuzzy	90	Inclinado (3°)	10	9	10	9.67
Fuzzy	135	Inclinado (3°)	20	19	20	19.67
Fuzzy	200	Inclinado (3°)	> 20	> 20	> 20	> 20

Fuente: Autor

#### 4.4 PRUEBAS DE DESPLAZAMIENTO CON EL PID DIFUSO

Previo a las pruebas de desplazamiento del robot se establece la conexión entre el computador que contiene el control remoto a través de la interfaz Tkinter y el Ballbot EV3, tal como se muestra en la Figura 4.16.

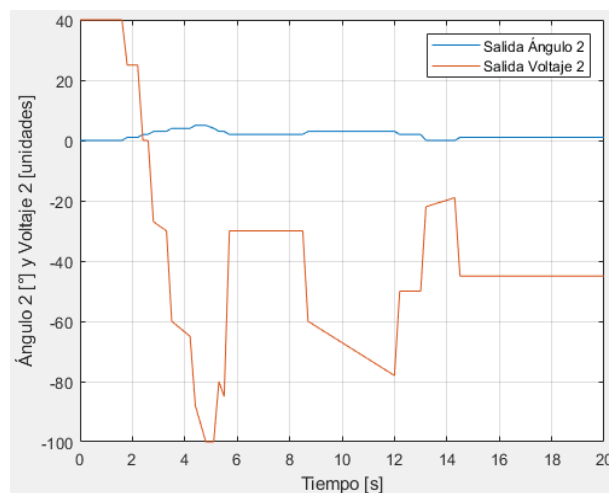
Figura 4.16. Conexión entre el control remoto y Ballbot EV3

```
Windows PowerShell
PS C:\Users\guill\Dropbox\proyecto Caap ballbot\remoteControl> python .\controlRemoto.py
Ingresar Ip del Robot: 192.168.1.104
Ip del Robot:192.168.1.104
connecting to 192.168.1.104 port 10000
```

Fuente: Autor

Una vez establecida la conexión, se procede con una serie de pruebas donde se tomaron lecturas del comportamiento de la variación del ángulo a medida que el robot se movía sobre el plano horizontal o inclinado. Durante la movilización, el valor de referencia del ángulo con respecto al eje vertical se modifica. Al evaluar el comportamiento de la Figura 4.17, se observa que para mantener el robot estable con un ángulo de referencia distinto de  $0^\circ$  el controlador debe ajustar el valor de voltaje del motor, inicialmente en el sentido del movimiento para inclinar el robot en esa dirección y después con el fin de mantener el equilibrio ajusta los motores en la dirección contraria provocando un desplazamiento.

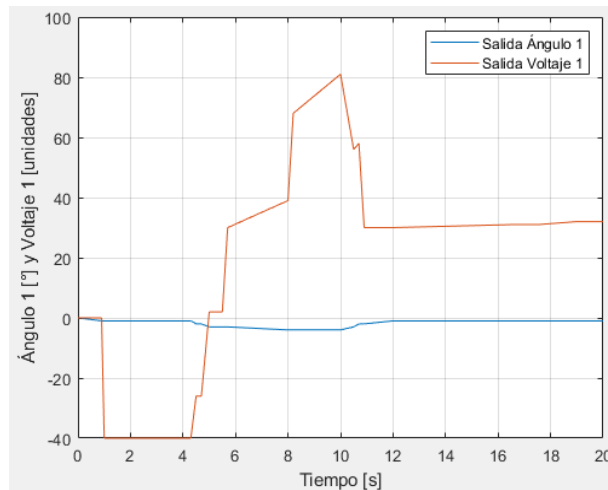
Figura 4.17. Desplazamiento del robot en el sentido izquierda derecha



Fuente: Autor

En la Figura 4.18 se observa el comportamiento para el desplazamiento en el sentido adelante-atrás, donde se detalla las curvas de la posición del ángulo 1 y la salida de voltaje 1. Debido a la inercia el robot presenta mayor movilidad de manera inicial y posteriormente esta se ajusta con el control de equilibrio. Cuando el voltaje 1 con respecto a la Figura se mantiene estable y el ángulo es diferente de  $0^\circ$  el robot presenta desplazamiento.

Figura 4.18. Desplazamiento del robot en el sentido adelante-atrás



Fuente: Autor

## 4.5 CONCLUSIONES

- El voltaje aplicado a los motores se define como la variable de entrada al Ballbot EV3, mientras que el ángulo de giro representa la variable de salida en el robot. La definición de los parámetros utilizados en el modelamiento del robot se basa en los criterios cinemáticos y dinámicos; estos fueron desarrollados con las ecuaciones de Lagrange para determinar las derivaciones de las ecuaciones de movimiento. Por último, el modelamiento y controlador del robot fueron elaborados mediante las herramientas de Simulink de Matlab.
- El controlador PID difuso fue elaborado en tres etapas, primero se empleó el bloque Tuner de Simulink para determinar las constantes PID del modelo propuesto, cuyos valores son de 55.5, 97.9 y 7.3. Luego, se aplicó la fuzzificación, las 8 reglas difusas y la defuzzificación para generar las condicionales a nuestro sistema, con el objetivo de minimizar el error del ángulo de giro. Por consiguiente, se realizó una calibración del controlador a fin de obtener un buen equilibrio y desplazamiento del Ballbot sobre la pelota, por lo que la ganancia proporcional se redujo en 35.5 unidades, la ganancia integral aumentó en 7.1 unidades y la ganancia derivativa disminuyó en 7 unidades.
- El control remoto se estableció mediante el protocolo TCP/IP por el puerto 10000, el cual es un servidor utilizado para la comunicación de redes entre el computador y el Ballbot EV3. Por lo que el robot, recibe datos (“a=adelante”, “r=retro”, “i=izquierda” y “d=derecha”) para el desplazamiento de este.

- Se han utilizado 4 pesas de diferentes pesos sobre el robot ante un plano inclinado a  $3^\circ$  con respecto a la horizontal, determinando que el robot puede mantener el equilibrio sin peso y con peso máximo de hasta 200 [gr] utilizando el controlador PID difuso. Sin embargo, cuando se realizó las pruebas con el controlador PID de Ziegler Nichols sucedió todo lo contrario, esto se debe al aumento de la constante integral (402.3), lo cual hace que el robot pierda estabilidad y se caiga.

#### **4.6 RECOMENDACIONES**

- Se recomienda que el ballbot EV3 no opere en planos inclinados mayores a  $3^\circ$  con respecto a la horizontal y su carga adicional no supere los 200 [gr] ya que el robot pierde el equilibrio.
- En caso de que las personas encargadas del laboratorio de electrónica de la Universidad Politécnica Salesiana Sede Quito deseen utilizar o realizar algún cambio en el software del Ballbot EV3 para trabajos futuros, se recomienda utilizar el editor de código denominado Visual Studio Code ya que facilita la instalación de las librerías pybricks EV3 de Lego Mindstorms.



## REFERENCIAS

- Bejar, P. (2018). Control avanzado aplicado a las funciones de un motor. Arequipa, Perú.
- Carvajal, J., Chen, G., & Ogmen, H. (2000). Fuzzy PID controller: Design, performance evaluation, and stability analysis. *Elsevier*, 249-270.
- Chengtao, C., Jiaxin, L., & Zuoyong, L. (2019). *Kinematic Analysis and Control Algorithm for the Ballbot*. Harbin: IEEE.
- Contaval. (2018). *Como afecta la inercia a los servos*. Recuperado el 29 de octubre de 2020, de <https://www.contaval.es/afecta-la-inercia-los-servos/>
- Corona, L., Abarca, G., & Mares, J. (2019). *Sensores y actuadores. Aplicaciones con Arduino* (segunda edición ed.). Grupo Editorial Patria.
- Cruz, H., & Gutiérrez, G. (2019). Rediseño de un robot móvil tipo ballbot. Ciudad de México, México.
- ElectricBricks. (2020). *Lego Center*. Obtenido de <https://www.electricbricks.com>
- Fankhauser, P. (2010). Modeling and control of a ballbot. Zúrich, Suiza.
- García, R. (2014). Desarrollo de un robot tipo ballbot para aplicaciones de control. Huajuapán de León, México.
- Golnaraghi, F. (2012). *MSE 312 Mechatronics Design II — Course Project*. Recuperado el 29 de octubre de 2020, de <https://www.accessengineeringlibrary.com/content/book/9781259643835/back-matter/appendix4#apxDlev1sec07>
- Hernández, R. (2010). *Introducción a los sistemas de control: conceptos, aplicaciones y simulación con MATLAB*. Atlacomulco: Prentice Hall.
- Johnson, J., Senthilnathan, R., Negi, M., Girishkumar, R., & Bhattacharjee, A. (2018). A fuzzy logic-in-loop control for a novel reduced height ballbot prototype. Kattanhulathur, India.
- LEGO. (2019). *Lego Education*. Obtenido de <https://education.lego.com/en-us>
- Lepuschitz, W., & Merdan, M. (2019). *Robotics in Education*. Suiza: Springer.
- Malý, J. (2018). Design and realization of the ball-riding robot for the promotion of the faculty. Praga, República Checa.
- Ninco. (2020). *Página Web Oficial de Ninco.Slot,Rc y más*. Obtenido de <https://www.ninco.com/>
- Ogata, K. (2010). *Ingeniería de control moderna*. Madrid: Prentice Hall.
- Pérez, H. (2015). *Física general* (primera edición ed.). Grupo editorial Patria S.A.
- Pérez, I., & León, B. (2007). *Lógica difusa para principiantes*. Caracas: Publicaciones UCAB.
- Philohome. (2020). *Lego 9V Technic Motors compared characteristics*. Recuperado el 29 de octubre de 2020, de <https://www.philohome.com/motors/motorcomp.htm>
- Quesada, A. (2017). Modelado y control de un robot omnidireccional en equilibrio sobre una rueda esférica. Costa Rica.
- Reyes, M. (2016). *Metodología de la Investigación*. Ciudad de México: Secretaría de Educación Pública.
- Román, J. (2018). Evaluación de proyectos fotovoltaicos usando Lógica Difusa. Cuernavaca, México: Universidad Autónoma del Estado de Morelos.
- Vaxasoftware. (2021). *Momentos de inercia de sólidos rígidos*. Recuperado el 14 de mayo de 2021, de [http://www.vaxasoftware.com/doc\\_edu/fis/mominercia.pdf](http://www.vaxasoftware.com/doc_edu/fis/mominercia.pdf)
- Villanes, L. (2016). Diseño de un algoritmo PID sintonizado mediante lógica fuzzy para controlar un brazo robótico. Lima, Perú.

Wang, J., Zhang, L., & Zeng, J. (2019). Modeling and controlling of spherical omnidirectional. Norte de China.

## ANEXOS

Anexo 1. Costos de los Ballbot comerciales .....	55
Anexo 2. Cálculo del Lagrangiano.....	56
Anexo 3. Programa principal del Ballbot EV3 .....	58
Anexo 4. Interfaz gráfica del control remoto .....	64
Anexo 5. Datos del ángulo de giro y voltaje usando diferentes controladores .....	66
Anexo 6. Funcionamiento del Ballbot EV3 .....	86
Anexo 7. Sintonización PID Ziegler Nichols en Lazo Abierto.....	88

## Anexo 1. Costos de los Ballbot comerciales

- Nanco

Inicio / Robótica / Robots R / C / Robot BallBot Naranja R / C



### Robot BallBot Naranja R / C

**29,99 €**

impuestos incluidos

edades


+ 6 Meses

cantidad

1


Añadir A La Cesta

- Lego Mindstorms




**LEGO Mindstorms 45514 Ev3 Cable Pack**  
Only 5 left in stock - order soon  
prime  
☐ This is a gift Learn more  
Qty: 1 ▼ Delete Save for later Compare with similar items

\$39.94




**LEGO EV3 Gyro Sensor**  
In Stock  
Shipped from: Ally U.S. Trading  
Gift options not available. Learn more  
Qty: 1 ▼ Delete Save for later

\$69.95



**Lego Mindstorms Ev3 Large Servo Motor**  
In Stock  
prime  
☐ This is a gift Learn more  
Qty: 1 ▼ Delete Save for later Compare with similar items

\$44.95



**LEGO Mindstorm Ev3 Core Set 45544 Ev3 Expansion Set 45560-LEGO Transformer/Charger 10VDC 45517 World-Wide Version(45517 Will Pack Inside 45560 or 45544 Carton Box)**  
In Stock  
Shipped from: Ally U.S. Trading  
Gift options not available. Learn more  
Qty: 1 ▼ Delete Save for later Compare with similar items

\$653.95

## Anexo 2. Cálculo del Lagrangiano

```
function [d2a1,d2a2] = fcn(a1,a2,da1,da2,Tm)
%r1=rr es el radio de la rueda
%r2=re es el radio de la pelota
%masas m1, m2 y m3 del cuerpo, pelota y rueda
%l distancia desde los centros de masas: robot y pelota
%unidades bajo el sistema SI, masa en kg y longitud en m
r1=0.052;
r2=0.21;
m1=0.7;
m2=0.1;
m3=0.25;
l=0.12;
d=0.096;
%momentos de inercia Ic cuerpo, Ie pelota y Ir rueda
Ic=0.003017;
Ie=0.002940;
Ir=0.000338;
%angulo y gravedad
r=r1 + r2;
alp=atan(d/(2*r));
g=9.8;
%calculos varios
s1=sin(a1);
c1=cos(a1);
v1=2*da2*s1*(r) + d*da2*c1;
v7=2*da2*s1*(r) - d*da2*c1;

v11=cos(a1)*(r);
v2=da2*r2 + da2*v11 - (d*da2*s1)/2;
v8=da2*r2 + da2*v11 + (d*da2*s1)/2;

v13=d*s1;
v4=2*v11 + v13;
v10=2*v11 - v13;

v12=sin(a1 + alp);
v3=da2^2*cos(a1 + alp)*v12*(r^2);
v14=sin(a1 - alp);
v9=da2^2*cos(a1 - alp)*v14*(r^2);

v15=da1^2*s1*r;
v5=da2^2*s1*r;
v6=(d*da2^2*c1)/2;

%aceleración a1 y a2
d2a2 = -((Tm - (m3*(v1*v2 - 2*v3))/2 + (m3*(v4*(v5 - v6)
+ v7*v8 - 4*v9))/2 - Tm*(r1/r2 + 1) + (m3*(v10*(v5 + v6)
+ v1*v2 - 4*v3))/2 - (m3*(v7*v8 - 2*v9))/2 + g*m3*v12*r +
```

$$\begin{aligned}
& g \cdot l \cdot m_1 \cdot s_1 + g \cdot m_3 \cdot v_{14} \cdot r) / (I_c + 4 \cdot I_r \cdot (r_2 / r_1 + 1)^2 + \\
& (m_3 \cdot (2 \cdot (v_{12} \cdot r)^2 + (v_{11} - v_{13} / 2) \cdot v_{10})) / 2 + (m_3 \cdot (v_4 \cdot (v_{11} + \\
& v_{13} / 2) + 2 \cdot (v_{14} \cdot r)^2)) / 2 + (m_1 \cdot (2 \cdot (l \cdot c_1)^2 + \\
& 2 \cdot (l \cdot s_1)^2)) / 2) - (m_3 \cdot r_2 \cdot (v_{15} - d \cdot da_1^2 \cdot c_1 / 2) + \\
& m_3 \cdot r_2 \cdot (v_{15} + d \cdot da_1^2 \cdot c_1 / 2) + (T_m \cdot r_2) / r_1 + \\
& da_1^2 \cdot l \cdot m_1 \cdot r_2 \cdot s_1) / (m_3 \cdot r_2 \cdot (v_{11} - v_{13} / 2) + m_3 \cdot r_2 \cdot (v_{11} + \\
& v_{13} / 2) - (4 \cdot r_2 \cdot I_r \cdot (r_2 / r_1 + 1)) / r_1 + l \cdot m_1 \cdot r_2 \cdot c_1) / ((I_e + \\
& m_1 \cdot r_2^2 + m_2 \cdot r_2^2 + 2 \cdot m_3 \cdot r_2^2 + \\
& (4 \cdot r_2^2 \cdot I_r) / r_1^2) / (m_3 \cdot r_2 \cdot (v_{11} - v_{13} / 2) + m_3 \cdot r_2 \cdot (v_{11} + \\
& v_{13} / 2) - (4 \cdot r_2 \cdot I_r \cdot (r_2 / r_1 + 1)) / r_1 + l \cdot m_1 \cdot r_2 \cdot c_1) - \\
& (m_3 \cdot r_2 \cdot v_4 / 2 + m_3 \cdot r_2 \cdot v_{10} / 2 - (4 \cdot r_2 \cdot I_r \cdot (r_2 / r_1 + 1)) / r_1 + \\
& l \cdot m_1 \cdot r_2 \cdot c_1) / (I_c + 4 \cdot I_r \cdot (r_2 / r_1 + 1)^2 + (m_3 \cdot (2 \cdot (v_{12} \cdot r)^2 + \\
& (v_{11} - v_{13} / 2) \cdot v_{10})) / 2 + (m_3 \cdot (v_4 \cdot (v_{11} + v_{13} / 2) + \\
& 2 \cdot (v_{14} \cdot r)^2)) / 2 + (m_1 \cdot (2 \cdot (l \cdot c_1)^2 + 2 \cdot (l \cdot s_1)^2)) / 2));
\end{aligned}$$

$$\begin{aligned}
d_{2a1} = & -(d_{2a2} \cdot I_e + d_{2a2} \cdot m_2 \cdot r_2^2 + m_1 \cdot r_2 \cdot (-l \cdot s_1 \cdot da_1^2 + \\
& d_{2a2} \cdot r_2) + m_3 \cdot r_2 \cdot (d_{2a2} \cdot r_2 - v_{15} + (d \cdot da_1^2 \cdot c_1) / 2) - \\
& m_3 \cdot r_2 \cdot (v_{15} - d_{2a2} \cdot r_2 + (d \cdot da_1^2 \cdot c_1) / 2) - (T_m \cdot r_2) / r_1 + \\
& (4 \cdot d_{2a2} \cdot r_2^2 \cdot I_r) / r_1^2) / (m_3 \cdot r_2 \cdot (v_{11} - v_{13} / 2) + m_3 \cdot r_2 \cdot (v_{11} \\
& + v_{13} / 2) - (4 \cdot r_2 \cdot I_r \cdot (r_2 / r_1 + 1)) / r_1 + l \cdot m_1 \cdot r_2 \cdot c_1);
\end{aligned}$$

### Anexo 3. Programa principal del Ballbot EV3

```
#!/usr/bin/env pybricks-micropython
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                  InfraredSensor, UltrasonicSensor, Gyro
                                  Sensor)
from pybricks.parameters import Port, Stop, Direction, Button, Color
from pybricks.tools import wait, StopWatch, DataLog
from pybricks.robotics import DriveBase
from pybricks.media.ev3dev import SoundFile, ImageFile
from threading import Thread
import socket

class fuzzyPid:
    def __init__(self,kp,ki,kd,Ts):
        self.error0=0
        self.derror0=0
        self.output=0
        self.Ts=Ts
        self.kp=kp-(Ts*ki/2)
        self.ki=ki*Ts
        self.kd=kd

    def calcular(self,error):
        ei= error*self.ki
        derror=error-self.error0
        ep= derror*self.kp
        ed= self.kd*(derror-self.derror0)
        self.error0=error
        self.derror0=derror
        L=5
        epp=(ep+L)/(2*L)
        eip=(ei+L)/(2*L)
        edp=(ed+L)/(2*L)
        if epp>1:
            epp=1
        if epp<0:
            epp=0
        if eip>1:
            eip=1
        if eip<0:
            eip=0
        if edp>1:
            edp=1
        if edp<0:
            edp=0
        do=[-120,-40, 0 ,40 ,120]
```

```

        u1=min(1-epp,1-eip,1-edp)
        u2=min(1-epp,1-eip,edp)
        u3=min(epp,1-eip,1-edp)
        u4=min(epp,1-eip,edp)
        u5=min(1-epp,eip,1-edp)
        u6=min(1-epp,eip,edp)
        u7=min(epp,eip,1-edp)
        u8=min(epp,eip,edp)
        if (u1+u2+u3+u4+u5+u6+u7+u8)>0.01:
            du=(u1*do[0]+u2*do[1]+u3*do[1]+u4*do[3]+u5*do[1]+u6*do[3]+u
7*do[3]+u8*do[4])/(u1+u2+u3+u4+u5+u6+u7+u8)
        else:
            du=0

        self.output+=du
        if self.output>100:
            self.output=100
        elif self.output<-100:
            self.output=-100

        return self.output

def encera(self):
    self.output = 0

class controladorPid:
    def __init__(self, Kp, Ki, Kd, Ts):
        self.Kp = Kp
        self.Ki = Ki
        self.Kd = Kd
        self.Ts = Ts
        self.accumulation_of_error = 0
        self.last_error = 0
        self.antiwindup = True
        self.limit = 100
        ev3.speaker.beep()
        self.derivativas=[]

    def calcular(self, error):
        proportional = error*self.Kp
        self.accumulation_of_error += error * self.Ts
        integral = self.accumulation_of_error * self.Ki
        if self.antiwindup:
            if integral > self.limit:
                integral = self.limit
                self.accumulation_of_error = integral/self.Ki
            elif integral < -self.limit:

```



```

        integral = -self.limit
        self.accumulation_of_error = integral/self.Ki

    derivativa = self.Kd*(error*0.05-0.05*self.last_error) /self.Ts

    self.last_error = error*0.05+ 0.95*self.last_error

    output = proportional+integral+derivativa
    return output

def encera(self):
    self.accumulation_of_error = 0

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_address = ('', 10000)
print('starting up on {} port {}'.format(*server_address))
sock.bind(server_address)
sock.listen(1)

adelante = False
retro = False
izquierda = False
derecha = False

servidorOn=True

def servidor():
    while servidorOn:
        print('waiting for a connection')
        connection, client_address = sock.accept()
        try:
            print('client connected:', client_address)
            while servidorOn:
                data = connection.recv(1)
                print('data recibido:{!r}'.format(data))
                if data:
                    if data == b'a':
                        global adelante
                        adelante = True
                    elif data == b'r':
                        global retro
                        retro = True
                    elif data == b'i':
                        global izquierda
                        izquierda = True
                    elif data == b'd':
                        global derecha
                        derecha = True

```

```

        elif data == b's':
            global adelante
            adelante = False
            global retro
            retro = False
            global izquierda
            izquierda = False
            global derecha
            derecha = False
            connection.send(data)
        else:
            break
    finally:
        connection.close()

    connection.close()

ev3 = EV3Brick()

t = Thread(target=servidor)
t.start()

motor_A = Motor(Port.A, Direction.CLOCKWISE)
motor_B = Motor(Port.B, Direction.CLOCKWISE)
motor_C = Motor(Port.C, Direction.CLOCKWISE)

gyro_sensor_1 = GyroSensor(Port.S1)
gyro_sensor_2 = GyroSensor(Port.S2)

TARGET_LOOP_PERIOD = 10 # ms
single_loop_timer = Stopwatch()

pid1 = fuzzyPid(25,105,0.3,0.01)
pid2 = fuzzyPid(25,105,0.3,0.01)

offset1=gyro_sensor_1.angle()
offset2=gyro_sensor_2.angle()

ev3.speaker.beep()
zonaMuerta = 0
factorDesplazamiento=2

memoria1=False
memoria1b=False
memoria2=False
memoria2b=False

iterations=0

```

```

while True:
    single_loop_timer.reset()
    angulo1 = gyro_sensor_1.angle()-offset1
    angulo2 = gyro_sensor_2.angle()-offset2
    referencia1=0
    referencia2=0
    maxMotor1=100
    maxMotor2=100

    if adelante:
        referencia1=-factorDesplazamiento
    elif retro:
        referencia1=factorDesplazamiento
    elif izquierda:
        referencia2=-factorDesplazamiento
    elif derecha:
        referencia2= factorDesplazamiento

    error1 =referencia1 -angulo1
    error2 =referencia2 -angulo2
    if (adelante or retro) and not memoria1:
        memoria1=True
        memoria1b=True

    if (not (adelante or retro) ) and memoria1:
        memoria1=False

    if (izquierda or derecha) and not memoria2:
        memoria2=True
        memoria2b=True
    if not (izquierda or derecha) and memoria2:
        memoria2=False

    # CALCULAR SALIDA PID
    o1 = pid1.calcular(error1)
    o2 = pid2.calcular(error2)

    if memoria1b:
        if adelante:
            if error1<0:
                maxMotor1=40
            else:
                memoria1b=False
        else:
            if error1>0:
                maxMotor1=40
            else:

```

```

        memoria1b=False

    if memoria2b:
        if izquierda:
            if error2<0:
                maxMotor2=40
            else:
                memoria2b=False
        else:
            if error2>0:
                maxMotor2=40
            else:
                memoria2b=False

    if o1>maxMotor1:
        o1=maxMotor1
    if o1<-maxMotor1:
        o1=-maxMotor1

    if o2>maxMotor2:
        o2=maxMotor2
    if o2<-maxMotor2:
        o2=-maxMotor2

    motor_A.dc(-o1)
    motor_B.dc(-o2)
    motor_C.dc(o2)
    if (abs(angulo1)>30 or abs(angulo2)>30):
        break

    wait(TARGET_LOOP_PERIOD - single_loop_timer.time())

motor_A.dc(0)
motor_B.dc(0)
motor_C.dc(0)
servidorOn=False

```

```
import tkinter
import socket

ventana = tkinter.Tk()

etiqueta = tkinter.Label(ventana, text="CONTROL REMOTO")
etiqueta.grid(row=0, column=0, columnspan=4)

envio=''

ipRobot=input('Ingresar Ip del Robot: ')
print('Ip del Robot:'+ipRobot)

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_address = (ipRobot, 10000)
print('connecting to {} port {}'.format(*server_address))
sock.connect(server_address)

def pressButton(accion):
    print(accion)
    global envio
    envio=accion
    if envio=='up':
        sock.sendall(b'a')
    elif envio=='down':
        sock.sendall(b'r')
    elif envio=='left':
        sock.sendall(b'i')
    elif envio=='righth':
        sock.sendall(b'd')

def releaseButton():
    print('release')
    global envio
    envio=''
    sock.sendall(b's')

def task():
    global envio
    if (envio):
        print(envio)
```

```

ventana.after(1000, task)

ventana.after(1000, task)

botonUp=tkinter.Button(ventana, text='Up')
botonUp.bind('<ButtonPress-1>', lambda event: pressButton('up'))
botonUp.bind('<ButtonRelease-1>', lambda event: releaseButton())
botonUp.grid(row=1, column=1,columnspan=2,
             sticky=tkinter.W+tkinter.E+tkinter.N+tkinter.S)

botonLeft=tkinter.Button(ventana, text='Left')
botonLeft.bind('<ButtonPress-1>', lambda event: pressButton('left'))
botonLeft.bind('<ButtonRelease-1>', lambda event: releaseButton())
botonLeft.grid(row=2, column=0,columnspan=2,
              sticky=tkinter.W+tkinter.E+tkinter.N+tkinter.S)

botonRigth=tkinter.Button(ventana, text='Rigth')
botonRigth.bind('<ButtonPress-1>', lambda event: pressButton('rigth'))
botonRigth.bind('<ButtonRelease-1>', lambda event: releaseButton())
botonRigth.grid(row=2, column=2,columnspan=2,
               sticky=tkinter.W+tkinter.E+tkinter.N+tkinter.S)

botonDown=tkinter.Button(ventana, text='Down')
botonDown.bind('<ButtonPress-1>', lambda event: pressButton('down'))
botonDown.bind('<ButtonRelease-1>', lambda event: releaseButton())
botonDown.grid(row=3, column=1,columnspan=2,
              sticky=tkinter.W+tkinter.E+tkinter.N+tkinter.S)

ventana.grid_columnconfigure(0, minsize=50)
ventana.grid_columnconfigure(1, minsize=50)
ventana.grid_columnconfigure(2, minsize=50)
ventana.grid_columnconfigure(3, minsize=50)

ventana.mainloop()

```

Anexo 5. Datos del ángulo de giro y voltaje usando diferentes controladores

- PID Ziegler Nichols

angulo1, salida1, angulo2, salida2	0, 0, 0, 0	-2, -100, -2, 100	-2, -100, -3, 100
0, 0, 0, 0	0, 0, -1, 100	-2, -100, -2, 100	-1, -100, -4, 100
0, 0, -1, 99.522000000000001	0, 0, -2, 100	-3, -100, -2, 100	0, 0, -3, 100
0, 0, -1, 100	0, 0, -2, 100	-3, -100, -3, 100	1, 100, -2, 100
0, 0, -1, 100	0, 0, -3, 100	-3, -100, -3, 100	2, 100, -1, 100
0, 0, -1, 100	0, 0, -3, 100	-3, -100, -3, 100	3, 100, 0, 0
0, 0, -1, 100	0, 0, -3, 100	-1, -100, -2, 100	3, 100, 0, 0
0, 0, -1, 100	0, 0, -2, 100	-1, -100, -1, 100	3, 100, 1, -100
0, 0, -1, 100	-1, -99.022000000000001, -2, 100	0, 0, 0, 0	3, 100, 2, -100
0, 0, 0, 0	-1, -100, -1, 100	0, 0, 0, 0	2, 100, 3, -100
0, 0, 1, -100	-1, -100, -1, 100	1, 100, 1, -100	1, 100, 3, -100
0, 0, 2, -100	-1, -100, 0, 0	2, 100, 1, -100	1, 100, 3, -100
0, 0, 4, -100	-1, -100, 1, -100	2, 100, 2, -100	1, 100, 3, -100
0, 0, 4, -100	-1, -100, 1, -100	1, 100, 2, -100	0, 0, 2, -100
0, 0, 4, -100	0, 0, 2, -100	1, 100, 2, -100	0, 0, 2, -100
0, 0, 3, -100	1, 100, 2, -100	1, 100, 2, -100	-1, -100, 2, -100
0, 0, 3, -100	2, 100, 2, -100	0, 0, 1, -100	-2, -100, 2, -100
0, 0, 2, -100	3, 100, 1, -100	-1, -100, 1, -100	-3, -100, 1, -100
0, 0, 2, -100	4, 100, 1, -100	-1, -100, 0, 0	-4, -100, 0, 0
0, 0, 2, -100	4, 100, 1, -100	-2, -100, 0, 0	-4, -100, 0, 0
0, 0, 2, -100	3, 100, 1, -100	-2, -100, -1, 100	-4, -100, -1, 100
0, 0, 1, -100	2, 100, 0, 0	-3, -100, -2, 100	-4, -100, -2, 100
0, 0, 0, 0	2, 100, -1, 100	-3, -100, -3, 100	-3, -100, -3, 100
0, 0, 0, 0	1, 100, -2, 100	-3, -100, -4, 100	-2, -100, -3, 100
0, 0, -1, 100	1, 100, -3, 100	-3, -100, -4, 100	-2, -100, -3, 100
0, 0, -2, 100	1, 100, -4, 100	-3, -100, -4, 100	-1, -100, -2, 100
0, 0, -2, 100	0, 0, -4, 100	-2, -100, -4, 100	0, 0, -1, 100
0, 0, -2, 100	-1, -100, -4, 100	-2, -100, -2, 100	1, 100, 0, 0
0, 0, -2, 100	-2, -100, -4, 100	-1, -100, -2, 100	3, 100, 2, -100
0, 0, -2, 100	-3, -100, -4, 100	0, 0, -1, 100	3, 100, 3, -100
0, 0, -1, 100	-4, -100, -3, 100	1, 100, -1, 100	4, 100, 4, -100
0, 0, -1, 100	-2, -100, -3, 100	1, 100, 0, 0	3, 100, 5, -100
0, 0, -1, 100	-2, -100, -2, 100	2, 100, 1, -100	3, 100, 5, -100
0, 0, 0, 0	-1, -100, -1, 100	2, 100, 2, -100	2, 100, 5, -100
0, 0, 1, -100	0, 0, 0, 0	2, 100, 2, -100	2, 100, 5, -100
0, 0, 2, -100	1, 100, 0, 0	2, 100, 3, -100	1, 100, 4, -100
0, 0, 3, -100	2, 100, 1, -100	1, 100, 3, -100	1, 100, 4, -100
0, 0, 3, -100	3, 100, 1, -100	0, 0, 3, -100	0, 0, 4, -100
0, 0, 3, -100	3, 100, 1, -100	0, 0, 3, -100	0, 0, 3, -100
0, 0, 3, -100	2, 100, 1, -100	-1, -100, 2, -100	-1, -100, 3, -100
0, 0, 2, -100	2, 100, 1, -100	-2, -100, 2, -100	-2, -100, 2, -100
0, 0, 2, -100	1, 100, 0, 0	-2, -100, 1, -100	-2, -100, 2, -100
0, 0, 2, -100	1, 100, 0, 0	-3, -100, 1, -100	-3, -100, 2, -100
0, 0, 2, -100	0, 0, 0, 0	-3, -100, 1, -100	-3, -100, 1, -100
0, 0, 2, -100	0, 0, 0, 0	-3, -100, 0, 0	-3, -100, 1, -100
0, 0, 1, -100	0, 0, -2, 100	-3, -100, -1, 100	-2, -100, 0, 0
0, 0, 0, 0	-1, -100, -2, 100	-2, -100, -1, 100	-1, -100, -1, 100

-1, -100, -1, 100	-2, -100, 66, -100	
0, 0, -2, 100	-3, -100, 65, -100	
1, 100, -2, 100	-4, -100, 65, -100	
3, 100, -1, 100	-4, -100, 62, -100	
3, 100, 0, 0	-5, -100, 63, -100	
3, 100, 1, -84.94859927095245	-5, -100, 64, -100	
2, 100, 3, -100	-6, -100, 64, -100	
2, 100, 4, -100	-6, -100, 64, -100	
1, 100, 5, -100	-6, -100, 64, -100	
1, 100, 7, -100	-5, -100, 63, -100	
1, 100, 7, -100	-4, -100, 63, -100	
0, 0, 8, -100	-3, -100, 63, -100	
0, 0, 8, -100	-2, -100, 59, -100	
-1, -100, 8, -100	-1, -91.96826019086703, 57, -100	
-2, -100, 9, -100	-1, -99.4714341717803, 56, -100	
-3, -100, 9, -100	-1, -100, 56, -100	
-4, -100, 9, -100	-2, -100, 52, -100	
-4, -100, 10, -100	-2, -100, 51, -100	
-3, -100, 10, -100	-3, -100, 51, -100	
-2, -100, 10, -100	-4, -100, 46, -100	
-2, -100, 11, -100	-5, -100, 44, -100	
-1, -100, 11, -100	-5, -100, 44, -100	
-1, -100, 13, -100	-4, -100, 44, -100	
-1, -100, 14, -100	-4, -100, 38, -100	
0, 0, 16, -100	-3, -100, 38, -100	
1, 100, 20, -100	-2, -100, 38, -100	
2, 100, 22, -100	-1, -100, 30, -100	
2, 100, 24, -100	-1, -100, 30, -100	
2, 100, 26, -100	0, 0, 29, -100	
1, 100, 28, -100	0, 0, 29, -100	
0, 0, 31, -100	0, 0, 29, -100	
-1, -99.69965708391731, 33, -100	0, 0, 29, -100	
-1, -100, 36, -100	1, 100, 29, -100	2, 100, 31, -100
-2, -100, 39, -100	1, 100, 29, -100	1, 100, 31, -100
-3, -100, 42, -100	1, 100, 29, -100	1, 100, 30, -100
-4, -100, 45, -100	1, 100, 29, -100	1, 100, 29, -100
-5, -100, 49, -100	1, 100, 29, -100	1, 100, 29, -100
-6, -100, 52, -100	1, 100, 29, -100	2, 100, 27, -100
-7, -100, 57, -100	1, 100, 30, -100	2, 100, 25, -100
-7, -100, 57, -100	1, 100, 30, -100	3, 100, 25, -100
-4, -100, 57, -100	2, 100, 30, -100	3, 100, 25, -100
-2, -100, 60, -100	2, 100, 31, -100	2, 100, 24, -100
-1, -100, 65, -100	2, 100, 31, -100	2, 100, 24, -100
-1, -100, 67, -100	2, 100, 31, -100	3, 100, 24, -100
-1, -100, 67, -100	2, 100, 31, -100	3, 100, 23, -100
-1, -100, 67, -100	2, 100, 31, -100	3, 100, 22, -100
-2, -100, 66, -100	2, 100, 31, -100	<b>3, 100, 22, -100</b>



- PID Fuzzy inicial

angulo1, salida1, angulo2, salida2	0, 0, -3, 100	0, 0, 2, -100
0, 0, 0, 0	0, 0, -2, 100	1, 100, 2, -100
-1, -100, 0, 0	-1, -100, -2, 100	1, 100, 2, -100
-1, -100, 0, 0	-1, -100, -2, 100	2, 100, 0, 0
-1, -100, 0, 0	-1, -100, -1, 66.31151257240368	2, 100, 0, 0
-1, -100, 0, 0	0, 0, 0, 0	2, 100, 0, 0
0, 0, 1, -100	0, 0, 1, -100	2, 100, -1, 100
3, 100, 1, -100	1, 73.10428294276085, 2, -100	1, 100, -1, 100
4, 100, 1, -100	1, 72.37075464848475, 2, -100	1, 100, -1, 100
5, 100, 1, -100	2, 100, 3, -100	1, 98.34962837888607, -1, 100
5, 100, 1, -100	2, 100, 3, -100	0, 0, -3, 100
4, 100, 1, -100	2, 100, 3, -100	0, 0, -3, 100
4, 100, 0, 0	1, 54.14030832487096, 3, -100	-1, -100, -3, 100
4, 100, 0, 0	1, 56.0863774923839, 3, -100	-1, -100, -3, 100
3, 100, -1, 100	0, 0, 3, -100	-1, -100, -3, 100
3, 100, -1, 100	0, 0, 3, -100	-3, -100, -3, 100
3, 100, -1, 100	-1, -100, 0, 0	-3, -100, -3, 100
3, 100, -1, 100	-1, -100, 0, 0	-4, -100, -2, 100
3, 100, -3, 100	-1, -100, 0, 0	-3, -100, -1, 29.68679400249469
2, 100, -3, 100	-2, -100, 0, 0	-3, -100, -1, 35.98831460224519
2, 100, -2, 100	-2, -100, 0, 0	-1, -56.14646719187926, 1, -100
2, 100, -2, 100	-2, -100, 0, 0	-1, -58.96882047269133, 1, -100
1, 9.377508716588252, -1, 77.19618507868159	-1, -100, -4, 100	0, 0, 1, -100
0, 0, -1, 77.47406657081342	-1, -100, -4, 100	0, 0, 1, -100
0, 0, -1, 77.82205991373203	0, 0, -4, 100	0, 0, 1, -100
0, 0, -1, 78.2331539223589	0, 0, -4, 100	0, 0, 1, -100
-1, -100, 0, 0	1, 100, -4, 100	1, 100, 1, -100
-1, -100, 1, -100	2, 100, -4, 100	0, 0, 1, -100
-1, -100, 1, -100	2, 100, -5, 100	0, 0, 1, -100
-1, -100, 3, -100	2, 100, -5, 100	0, 0, 1, -100
-1, -100, 3, -100	2, 100, -5, 100	0, 0, -2, 100
0, 0, 3, -100	2, 100, -4, 100	-1, -99.70219638443904, -2, 100
1, 94.28536637307124, 3, -100	2, 100, -4, 100	-1, -96.30887674599515, -3, 100
1, 91.43372973576406, 3, -100	1, 80.81295993238383, -3, 100	-1, -93.35278907139566, -3, 100
3, 100, 3, -100	1, 80.58126393914552, -2, 100	-2, -100, -3, 100
3, 100, 1, -65.57303885024095	0, 0, -2, 100	-2, -100, -3, 100
4, 100, 1, -67.60063496521688	0, 0, -2, 100	-1, -74.78011023304737, -3, 100
4, 100, 1, -69.52337146869517	0, 0, -2, 100	-1, -74.56429920974266, -8, 100
4, 100, 0, 0	-2, -100, -2, 100	0, 0, -7, 100
4, 100, 0, 0	-2, -100, -2, 100	1, 100, -6, 100
4, 100, 0, 0	-2, -100, -2, 100	1, 100, -5, 100
3, 100, -2, 100	-2, -100, -1, 19.81801205095323	1, 100, -5, 100
3, 100, -2, 100	-2, -100, 1, -100	1, 100, -5, 100
3, 100, -2, 100	-2, -100, 2, -100	0, 0, -6, 100
1, -13.86499211638203, -2, 100	-2, -100, 2, -100	0, 0, -6, 100
1, -2.279292904743798, -2, 100	-1, -88.55051044540279, 2, -100	-1, -100, -5, 100
1, 8.245736385730467, -3, 100	0, 0, 2, -100	-1, -100, -5, 100

-2, -100, -5, 100	0, 0, -6, 100	
-3, -100, -5, 100	1, 100, -7, 100	
-3, -100, -4, 100	2, 100, -7, 100	
-2, -100, -4, 100	3, 100, -9, 100	
-2, -100, -3, 100	4, 100, -9, 100	
0, 0, -3, 100	5, 100, -9, 100	
0, 0, -3, 100	4, 100, -11, 100	
1, 100, -3, 100	4, 100, -11, 100	
1, 100, -3, 100	4, 100, -11, 100	
2, 100, -2, 83.98611222737269	4, 100, -11, 100	
2, 100, -2, 96.68750100463519	3, 100, -11, 100	3, 100, -8, -100
2, 100, -2, 100	3, 100, -11, 100	2, 100, -7, -100
1, 100, -2, 100	3, 100, -11, 100	2, 100, -6, -100
0, 0, -3, 100	3, 100, -11, 100	1, 51.74151061605524, -5, -100
0, 0, -3, 100	3, 100, -20, 100	1, 55.00435955444966, -5, -100
0, 0, -3, 100	3, 100, -20, 100	0, 0, -4, -100
-1, -100, -3, 100	3, 100, -20, 100	0, 0, -4, -100
-1, -100, -3, 100	3, 100, -20, 100	-205
-1, -100, -2, 100	1, -5.64558816464627, -20, 100	-204
-2, -100, -2, 100	1, 6.58667065181848, -20, 100	-204
-2, -100, -2, 100	1, 17.69360358663659, -24, 100	-204
-2, -100, -2, 100	1, 27.78774322797291, -25, 100	0, 0, -3, -100
-2, -100, -2, 100	1, 36.97036890517563, -25, 100	0, 0, -4, 35.20268537740805
-2, -100, -1, 30.60116959175446	0, 0, -25, 100	0, 0, -4, 63.88241683966715
-1, -82.0896360354218, -1, 43.09105263257902	0, 0, -25, 100	1, 100, -4, 89.69417515570044
0, 0, 0, 0	1, 16.02501193187305, -25, 100	2, 100, -4, 100
1, 100, 0, 0	2, 100, -25, 100	2, 100, -4, 100
2, 100, 0, 0	2, 100, -25, 100	3, 100, -4, 100
2, 100, 0, 0	2, 100, -25, 100	2, 100, -3, 41.08414831965468
3, 100, 0, 0	2, 100, -27, 100	2, 100, -3, 63.62573348768956
3, 100, 0, 0	2, 100, -27, 100	2, 100, -2, -44.58683986107923
3, 100, 0, 0	2, 100, -27, 100	1, 56.87265222174254, -1, -100
2, 100, 0, 0	1, 15.39132436437799, -27, 100	0, 0, 0, 0
2, 100, -2, 100	0, 0, -26, 100	0, 0, 1, -100
2, 100, -3, 100	-1, -100, -26, 100	0, 0, 1, -100
1, 68.4821532050285, -3, 100	-1, -100, -25, 100	0, 0, 2, -100
1, 70.16883788452566, -3, 100	-1, -100, -24, 100	0, 0, 2, -100
0, 0, -4, 100	-2, -100, -24, 100	0, 0, 3, -100
-1, -100, -4, 100	-2, -100, -23, 100	-1, -100, 3, -100
-1, -100, -4, 100	-2, -100, -21, 100	-1, -100, 4, -100
-1, -100, -4, 100	-1, -100, -20, 100	-1, -100, 4, -100
-2, -100, -4, 100	-1, -100, -18, 100	-1, -100, 4, -100
-2, -100, -6, 100	0, 0, -17, 100	0, 0, 4, -100
-2, -100, -6, 100	1, 100, -15, 100	1, 100, 5, -100
-2, -100, -6, 100	2, 100, -14, 100	2, 100, 5, -100
-1, -100, -6, 100	3, 100, -12, 100	3, 100, 5, -100
-1, -100, -6, 100	3, 100, -10, -49.51296522933695	3, 100, 6, -100
-1, -100, -6, 100	3, 100, -9, -100	2, 100, 7, -100
		1, 69.94395097510831, 7, -100

- PID Fuzzy calibrado

angulo1, salida1, angulo2, salida2	0, 0, 1, -48.25910931174089
0, 0, 0, 0	1, 41.02564102564103, 0, 0
0, 0, -1, 41.02564102564103	1, 42.05128205128206, 0, 0
0, 0, -1, 42.05128205128206	1, 43.07692307692309, 0, 0
0, 0, -1, 43.07692307692309	1, 44.10256410256411, -1, 41.02564102564103
0, 0, -1, 44.10256410256411	1, 45.12820512820515, -2, 83.13090418353577
0, 0, -1, 45.12820512820515	1, 46.15384615384618, -2, 85.23616734143051
0, 0, -1, 46.15384615384618	1, 47.17948717948721, -2, 87.34143049932525
0, 0, 0, 0	1, 48.20512820512824, -2, 89.44669365721998
0, 0, 0, 0	0, 0, -2, 91.55195681511472
0, 0, 0, 0	0, 0, -2, 93.65721997300946
1, 41.02564102564103, 1, -41.02564102564102	0, 0, -1, 54.68286099865049
1, 42.05128205128206, 1, -42.05128205128204	0, 0, 0, 0
1, 43.07692307692309, 1, -43.07692307692307	0, 0, 1, -41.02564102564102
1, 44.10256410256411, 0, 0	0, 0, 1, -42.05128205128204
1, 45.12820512820515, 0, 0	0, 0, 1, -43.07692307692307
1, 46.15384615384618, 0, 0	1, 41.02564102564103, 1, -44.10256410256409
1, 47.17948717948721, 0, 0	1, 42.05128205128206, 0, 0
1, 48.20512820512824, 0, 0	1, 43.07692307692309, 0, 0
1, 49.23076923076927, 0, 0	1, 44.10256410256411, 0, 0
1, 50.25641025641029, 0, 0	1, 45.12820512820515, -1, 41.02564102564103
1, 51.28205128205132, 0, 0	1, 46.15384615384618, -1, 42.05128205128206
0, 0, 0, 0	1, 47.17948717948721, -1, 43.07692307692309
0, 0, 0, 0	1, 48.20512820512824, -1, 44.10256410256411
0, 0, 0, 0	1, 49.23076923076927, -1, 45.12820512820515
0, 0, 0, 0	1, 50.25641025641029, -1, 46.15384615384618
-1, -41.02564102564102, 0, 0	0, 0, -1, 47.17948717948721
0, 0, 0, 0	0, 0, -2, 89.28475033738195
0, 0, 0, 0	0, 0, -1, 50.31039136302297
0, 0, 0, 0	-1, -41.02564102564102, -1, 51.336032388664
0, 0, 0, 0	-1, -42.05128205128204, 0, 0
0, 0, -1, 41.02564102564103	0, 0, 0, 0
0, 0, -1, 42.05128205128206	0, 0, 0, 0
1, 41.02564102564103, -1, 43.07692307692309	0, 0, 1, -41.02564102564102
1, 42.05128205128206, -1, 44.10256410256411	1, 41.02564102564103, 1, -42.05128205128204
1, 43.07692307692309, -1, 45.12820512820515	1, 42.05128205128206, 1, -43.07692307692307
1, 44.10256410256411, -1, 46.15384615384618	1, 43.07692307692309, 0, 0
1, 45.12820512820515, -1, 47.17948717948721	1, 44.10256410256411, 0, 0
1, 46.15384615384618, -1, 48.20512820512824	1, 45.12820512820515, 0, 0
1, 47.17948717948721, 0, 0	1, 46.15384615384618, 0, 0
1, 48.20512820512824, 0, 0	1, 47.17948717948721, 0, 0
0, 0, 1, -41.02564102564102	1, 48.20512820512824, 0, 0
0, 0, 1, -42.05128205128204	1, 49.23076923076927, 0, 0
0, 0, 1, -43.07692307692307	1, 50.25641025641029, 0, 0
0, 0, 2, -85.18218623481781	0, 0, 0, 0
0, 0, 1, -46.20782726045884	0, 0, 0, 0
0, 0, 1, -47.23346828609986	0, 0, 0, 0

0, 0, 0, 0	0, 0, -1, 36.43724696356283
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
1, 41.02564102564103, 0, 0	-1, -41.02564102564102, 0, 0
1, 42.05128205128206, 0, 0	-1, -42.05128205128204, 0, 0
1, 43.07692307692309, 0, 0	-1, -43.07692307692307, 0, 0
1, 44.10256410256411, 0, 0	-1, -44.10256410256409, 0, 0
1, 45.12820512820515, 1, -41.02564102564102	0, 0, 0, 0
0, 0, 1, -42.05128205128204	0, 0, -1, 41.02564102564103
0, 0, 1, -43.07692307692307	0, 0, -1, 42.05128205128206
-1, -41.02564102564102, 1, -44.10256410256409	0, 0, -1, 43.07692307692309
-1, -42.05128205128204, 1, -45.12820512820511	0, 0, -1, 44.10256410256411
-1, -43.07692307692307, 1, -46.15384615384613	0, 0, -1, 45.12820512820515
0, 0, 1, -47.17948717948715	0, 0, 0, 0
0, 0, 1, -48.20512820512818	0, 0, 0, 0
0, 0, -1, -7.179487179487147	0, 0, 1, -41.02564102564102
0, 0, -1, -6.153846153846121	0, 0, 1, -42.05128205128204
0, 0, -1, -5.128205128205096	0, 0, 1, -43.07692307692307
0, 0, -1, -4.10256410256407	0, 0, 1, -44.10256410256409
0, 0, -1, -3.076923076923045	0, 0, 0, 0
0, 0, -1, -2.051282051282019	0, 0, 0, 0
-1, -41.02564102564102, -2, 40.05398110661273	0, 0, 1, -41.02564102564102
-1, -42.05128205128204, -2, 42.15924426450747	0, 0, 1, -42.05128205128204
-1, -43.07692307692307, -1, 3.184885290148486	1, 41.02564102564103, 1, -43.07692307692307
-1, -44.10256410256409, -1, 4.210526315789512	1, 42.05128205128206, 1, -44.10256410256409
0, 0, -1, 5.236167341430537	1, 43.07692307692309, 0, 0
0, 0, -1, 6.261808367071563	1, 44.10256410256411, 0, 0
0, 0, -2, 48.36707152496631	1, 45.12820512820515, 0, 0
0, 0, -2, 50.47233468286105	1, 46.15384615384618, 0, 0
0, 0, -2, 52.57759784075579	1, 47.17948717948721, 1, -41.02564102564102
0, 0, -2, 54.68286099865053	1, 48.20512820512824, 1, -42.05128205128204
0, 0, -2, 56.78812415654527	0, 0, 1, -43.07692307692307
0, 0, -2, 58.89338731444001	0, 0, 1, -44.10256410256409
-1, -41.02564102564102, -2, 60.99865047233475	0, 0, 1, -45.12820512820511
-1, -42.05128205128204, -2, 63.10391363022949	0, 0, 1, -46.15384615384613
-1, -43.07692307692307, -1, 24.12955465587051	0, 0, 1, -47.17948717948715
-1, -44.10256410256409, -1, 25.15519568151154	0, 0, 1, -48.20512820512818
-1, -45.12820512820511, -1, 26.18083670715256	0, 0, 1, -49.2307692307692
-1, -46.15384615384613, -1, 27.20647773279358	0, 0, 1, -50.25641025641022
0, 0, -1, 28.23211875843461	0, 0, 1, -51.28205128205124
0, 0, -1, 29.25775978407564	0, 0, 1, -52.30769230769226
1, 41.02564102564103, -1, 30.28340080971666	0, 0, 1, -53.33333333333329
1, 42.05128205128206, -1, 31.30904183535769	1, 41.02564102564103, 1, -54.35897435897431
1, 43.07692307692309, -1, 32.33468286099871	1, 42.05128205128206, 1, -55.38461538461533
1, 44.10256410256411, -1, 33.36032388663974	1, 43.07692307692309, 1, -56.41025641025635
1, 45.12820512820515, -1, 34.38596491228077	1, 44.10256410256411, 1, -57.43589743589737
0, 0, -1, 35.4116059379218	1, 45.12820512820515, 1, -58.4615384615384



1, 46.15384615384618, 0, 0	0, 0, -2, 91.49797570850206
0, 0, 0, 0	0, 0, -1, 52.52361673414308
0, 0, -1, 41.02564102564103	0, 0, -1, 53.54925775978411
0, 0, -1, 42.05128205128206	0, 0, -1, 54.57489878542514
0, 0, -1, 43.07692307692309	0, 0, 0, 0
0, 0, -1, 44.10256410256411	0, 0, 1, -41.02564102564102
0, 0, 0, 0	0, 0, 1, -42.05128205128204
0, 0, 0, 0	0, 0, 1, -43.07692307692307
0, 0, 0, 0	0, 0, 1, -44.10256410256409
0, 0, 0, 0	-1, -41.02564102564102, 1, -45.12820512820511
0, 0, 0, 0	-1, -42.05128205128204, 0, 0
0, 0, 0, 0	-1, -43.07692307692307, 0, 0
0, 0, 0, 0	-1, -44.10256410256409, 0, 0
0, 0, 0, 0	-1, -45.12820512820511, 0, 0
0, 0, 0, 0	-1, -46.15384615384613, 0, 0
0, 0, 0, 0	-1, -47.17948717948715, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	1, 41.02564102564103, 0, 0
0, 0, 0, 0	1, 42.05128205128206, -1, 41.02564102564103
0, 0, 0, 0	1, 43.07692307692309, -1, 42.05128205128206
0, 0, 0, 0	1, 44.10256410256411, 0, 0
0, 0, 0, 0	1, 45.12820512820515, 0, 0
0, 0, 0, 0	0, 0, -1, 41.02564102564103
0, 0, -1, 41.02564102564103	1, 41.02564102564103, -1, 42.05128205128206
0, 0, -1, 42.05128205128206	1, 42.05128205128206, -1, 43.07692307692309
0, 0, -1, 43.07692307692309	1, 43.07692307692309, 0, 0
0, 0, -1, 44.10256410256411	0, 0, 0, 0
0, 0, -1, 45.12820512820515	0, 0, 0, 0
0, 0, -1, 46.15384615384618	-1, -41.02564102564102, 1, -41.02564102564102
0, 0, -1, 47.17948717948721	-1, -42.05128205128204, 1, -42.05128205128204
-1, -41.02564102564102, -1, 48.20512820512824	-1, -43.07692307692307, 1, -43.07692307692307
-1, -42.05128205128204, -1, 49.23076923076927	-1, -44.10256410256409, 1, -44.10256410256409
-1, -43.07692307692307, -1, 50.25641025641029	-1, -45.12820512820511, 1, -45.12820512820511
-1, -44.10256410256409, 0, 0	0, 0, 0, 0
-1, -45.12820512820511, 0, 0	0, 0, 1, -41.02564102564102
-1, -46.15384615384613, 0, 0	0, 0, 1, -42.05128205128204
-1, -47.17948717948715, 0, 0	1, 41.02564102564103, 1, -43.07692307692307
-1, -48.20512820512818, 0, 0	1, 42.05128205128206, 1, -44.10256410256409
0, 0, 0, 0	1, 43.07692307692309, 1, -45.12820512820511
0, 0, 0, 0	1, 44.10256410256411, 0, 0
1, 41.02564102564103, -1, 41.02564102564103	0, 0, 0, 0
1, 42.05128205128206, -1, 42.05128205128206	0, 0, 0, 0
1, 43.07692307692309, -1, 43.07692307692309	0, 0, 0, 0
1, 44.10256410256411, -2, 85.18218623481784	0, 0, 0, 0
0, 0, -2, 87.28744939271257	0, 0, 0, 0
0, 0, -2, 89.39271255060731	0, 0, 1, -41.02564102564102

0, 0, 1, -42.05128205128204	0, 0, 0, 0
0, 0, 1, -43.07692307692307	0, 0, 0, 0
1, 41.02564102564103, 1, -44.10256410256409	1, 41.02564102564103, -1, 41.02564102564103
1, 42.05128205128206, 1, -45.12820512820511	1, 42.05128205128206, -1, 42.05128205128206
1, 43.07692307692309, 1, -46.15384615384613	0, 0, -1, 43.07692307692309
1, 44.10256410256411, 1, -47.17948717948715	0, 0, -1, 44.10256410256411
1, 45.12820512820515, 1, -48.20512820512818	0, 0, -1, 45.12820512820515
1, 46.15384615384618, 1, -49.2307692307692	0, 0, -1, 46.15384615384618
0, 0, 1, -50.25641025641022	0, 0, -1, 47.17948717948721
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 1, -41.02564102564102
0, 0, 0, 0	0, 0, 1, -42.05128205128204
1, 41.02564102564103, 0, 0	-1, -41.02564102564102, 1, -43.07692307692307
1, 42.05128205128206, 0, 0	-1, -42.05128205128204, 1, -44.10256410256409
1, 43.07692307692309, 0, 0	-1, -43.07692307692307, 1, -45.12820512820511
1, 44.10256410256411, 0, 0	-1, -44.10256410256409, 1, -46.15384615384613
1, 45.12820512820515, 0, 0	0, 0, 0, 0
1, 46.15384615384618, 0, 0	0, 0, 0, 0
1, 47.17948717948721, 0, 0	1, 41.02564102564103, 0, 0
0, 0, 0, 0	1, 42.05128205128206, 0, 0
0, 0, 0, 0	1, 43.07692307692309, -1, 41.02564102564103
0, 0, 0, 0	1, 44.10256410256411, -1, 42.05128205128206
0, 0, 0, 0	1, 45.12820512820515, -1, 43.07692307692309
-1, -41.02564102564102, 0, 0	0, 0, 0, 0
0, 0, 0, 0	1, 41.02564102564103, 0, 0
0, 0, 0, 0	1, 42.05128205128206, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, -1, 41.02564102564103	0, 0, 0, 0
0, 0, -1, 42.05128205128206	0, 0, 0, 0
0, 0, -1, 43.07692307692309	0, 0, 0, 0
0, 0, -1, 44.10256410256411	0, 0, 0, 0
0, 0, -1, 45.12820512820515	0, 0, 1, -41.02564102564102
0, 0, -1, 46.15384615384618	0, 0, 1, -42.05128205128204
0, 0, -1, 47.17948717948721	1, 41.02564102564103, 1, -43.07692307692307
0, 0, -1, 48.20512820512824	1, 42.05128205128206, 0, 0
0, 0, -1, 49.23076923076927	1, 43.07692307692309, 1, -41.02564102564102
0, 0, -1, 50.25641025641029	0, 0, 1, -42.05128205128204
-1, -41.02564102564102, 0, 0	0, 0, 1, -43.07692307692307
-1, -42.05128205128204, 0, 0	0, 0, 1, -44.10256410256409
-1, -43.07692307692307, 0, 0	0, 0, 1, -45.12820512820511
-1, -44.10256410256409, 0, 0	0, 0, 1, -46.15384615384613
-1, -45.12820512820511, 0, 0	0, 0, 1, -47.17948717948715
-1, -46.15384615384613, 0, 0	1, 41.02564102564103, 0, 0
-1, -47.17948717948715, 0, 0	1, 42.05128205128206, 0, 0
0, 0, 0, 0	1, 43.07692307692309, 0, 0

1, 44.10256410256411, 0, 0	0, 0, 0, 0
1, 45.12820512820515, 0, 0	0, 0, 0, 0
1, 46.15384615384618, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 1, -41.02564102564102
0, 0, 0, 0	0, 0, 1, -42.05128205128204
0, 0, 0, 0	0, 0, 1, -43.07692307692307
0, 0, 0, 0	0, 0, 1, -44.10256410256409
0, 0, 0, 0	0, 0, 1, -45.12820512820511
0, 0, 0, 0	1, 41.02564102564103, 1, -46.15384615384613
0, 0, 0, 0	1, 42.05128205128206, 1, -47.17948717948715
0, 0, 0, 0	1, 43.07692307692309, 0, 0
0, 0, 0, 0	1, 44.10256410256411, 0, 0
0, 0, 0, 0	1, 45.12820512820515, 0, 0
0, 0, 0, 0	1, 46.15384615384618, 0, 0
0, 0, 0, 0	1, 47.17948717948721, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 1, -41.02564102564102
0, 0, 0, 0	0, 0, 1, -42.05128205128204
-1, -41.02564102564102, 0, 0	1, 41.02564102564103, 1, -43.07692307692307
-1, -42.05128205128204, -1, 41.02564102564103	1, 42.05128205128206, 1, -44.10256410256409
-1, -43.07692307692307, -1, 42.05128205128206	1, 43.07692307692309, 1, -45.12820512820511
-1, -44.10256410256409, -1, 43.07692307692309	1, 44.10256410256411, 1, -46.15384615384613
-1, -45.12820512820511, -1, 44.10256410256411	1, 45.12820512820515, 1, -47.17948717948715
-1, -46.15384615384613, -1, 45.12820512820515	1, 46.15384615384618, 1, -48.20512820512818
-1, -47.17948717948715, -1, 46.15384615384618	1, 47.17948717948721, 1, -49.2307692307692
-1, -48.20512820512818, -1, 47.17948717948721	1, 48.20512820512824, 1, -50.25641025641022
-1, -49.2307692307692, -1, 48.20512820512824	1, 49.23076923076927, 0, 0
0, 0, -1, 49.23076923076927	0, 0, 0, 0
0, 0, -1, 50.25641025641029	0, 0, -1, 41.02564102564103
1, 41.02564102564103, 0, 0	0, 0, -1, 42.05128205128206
1, 42.05128205128206, 0, 0	0, 0, -1, 43.07692307692309
1, 43.07692307692309, 0, 0	-1, -41.02564102564102, -1, 44.10256410256411
0, 0, 0, 0	0, 0, -1, 45.12820512820515
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, -1, 41.02564102564103
0, 0, 0, 0	0, 0, -1, 42.05128205128206
0, 0, -1, 41.02564102564103	0, 0, -1, 43.07692307692309
0, 0, -1, 42.05128205128206	0, 0, 0, 0
0, 0, -1, 43.07692307692309	-1, -41.02564102564102, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 1, -41.02564102564102
0, 0, 0, 0	0, 0, 1, -42.05128205128204
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0

-1, -41.02564102564102, 0, 0	0, 0, -1, 41.02564102564103
-1, -42.05128205128204, 0, 0	0, 0, -1, 42.05128205128206
-1, -43.07692307692307, 0, 0	0, 0, -1, 43.07692307692309
-1, -44.10256410256409, 0, 0	0, 0, 0, 0
-1, -45.12820512820511, 0, 0	1, 41.02564102564103, 0, 0
0, 0, 0, 0	1, 42.05128205128206, 0, 0
0, 0, 0, 0	1, 43.07692307692309, 0, 0
0, 0, 0, 0	1, 44.10256410256411, 0, 0
0, 0, 0, 0	1, 45.12820512820515, 0, 0
1, 41.02564102564103, 0, 0	1, 46.15384615384618, 0, 0
0, 0, -1, 41.02564102564103	1, 47.17948717948721, 0, 0
0, 0, -1, 42.05128205128206	0, 0, 0, 0
0, 0, -1, 43.07692307692309	0, 0, 0, 0
0, 0, -1, 44.10256410256411	0, 0, 0, 0
0, 0, -1, 45.12820512820515	0, 0, 0, 0
0, 0, -1, 46.15384615384618	0, 0, 0, 0
0, 0, -1, 47.17948717948721	0, 0, 0, 0
0, 0, -1, 48.20512820512824	0, 0, 0, 0
-1, -41.02564102564102, -1, 49.23076923076927	0, 0, 0, 0
-1, -42.05128205128204, -1, 50.25641025641029	0, 0, 0, 0
-1, -43.07692307692307, -1, 51.28205128205132	0, 0, 0, 0
-1, -44.10256410256409, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -41.02564102564102	0, 0, 0, 0
0, 0, 1, -42.05128205128204	0, 0, 0, 0
0, 0, 1, -43.07692307692307	0, 0, 0, 0
0, 0, 1, -44.10256410256409	0, 0, 0, 0
0, 0, 0, 0	-1, -41.02564102564102, 0, 0
0, 0, 0, 0	-1, -42.05128205128204, 0, 0
0, 0, 0, 0	-1, -43.07692307692307, 0, 0
0, 0, 0, 0	-1, -44.10256410256409, 0, 0
0, 0, 0, 0	-1, -45.12820512820511, 0, 0
0, 0, 0, 0	-1, -46.15384615384613, 0, 0
0, 0, 0, 0	-1, -47.17948717948715, 0, 0
0, 0, 0, 0	-1, -48.20512820512818, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
1, 41.02564102564103, 1, -41.02564102564102	1, 41.02564102564103, 0, 0
1, 42.05128205128206, 1, -42.05128205128204	1, 42.05128205128206, 0, 0
1, 43.07692307692309, 1, -43.07692307692307	1, 43.07692307692309, 0, 0
1, 44.10256410256411, 1, -44.10256410256409	0, 0, 0, 0
1, 45.12820512820515, 1, -45.12820512820511	0, 0, 0, 0
1, 46.15384615384618, 1, -46.15384615384613	0, 0, 0, 0
1, 47.17948717948721, 1, -47.17948717948715	0, 0, 0, 0
1, 48.20512820512824, 0, 0	0, 0, 0, 0
1, 49.23076923076927, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0



0, 0, 0, 0	0, 0, 1, -41.02564102564102
0, 0, 0, 0	0, 0, 1, -42.05128205128204
-1, -41.02564102564102, 0, 0	0, 0, 1, -43.07692307692307
-1, -42.05128205128204, 0, 0	1, 41.02564102564103, 1, -44.10256410256409
-1, -43.07692307692307, 0, 0	1, 42.05128205128206, 1, -45.12820512820511
-1, -44.10256410256409, 0, 0	1, 43.07692307692309, 1, -46.15384615384613
-1, -45.12820512820511, 0, 0	0, 0, 1, -47.17948717948715
-1, -46.15384615384613, 0, 0	0, 0, 1, -48.20512820512818
-1, -47.17948717948715, 0, 0	0, 0, 1, -49.2307692307692
-1, -48.20512820512818, 0, 0	0, 0, 1, -50.25641025641022
0, 0, -1, 41.02564102564103	0, 0, 1, -51.28205128205124
1, 41.02564102564103, -1, 42.05128205128206	0, 0, 0, 0
1, 42.05128205128206, -1, 43.07692307692309	0, 0, 0, 0
2, 84.15654520917681, -1, 44.10256410256411	0, 0, 0, 0
2, 86.26180836707155, -1, 45.12820512820515	-1, -41.02564102564102, 0, 0
1, 47.28744939271256, -1, 46.15384615384618	-1, -42.05128205128204, 0, 0
1, 48.31309041835359, -1, 47.17948717948721	-1, -43.07692307692307, 0, 0
1, 49.33873144399462, 0, 0	-1, -44.10256410256409, 0, 0
1, 50.36437246963565, 0, 0	-1, -45.12820512820511, 0, 0
0, 0, 0, 0	-1, -46.15384615384613, 0, 0
0, 0, 0, 0	-1, -47.17948717948715, 0, 0
0, 0, 0, 0	-1, -48.20512820512818, 0, 0
-1, -41.02564102564102, 0, 0	-1, -49.2307692307692, 0, 0
-1, -42.05128205128204, 0, 0	0, 0, 0, 0
-1, -43.07692307692307, 0, 0	1, 41.02564102564103, 0, 0
0, 0, 0, 0	1, 42.05128205128206, 0, 0
0, 0, 0, 0	1, 43.07692307692309, 0, 0
1, 41.02564102564103, 0, 0	1, 44.10256410256411, 0, 0
1, 42.05128205128206, 0, 0	1, 45.12820512820515, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
1, 41.02564102564103, 0, 0	0, 0, 0, 0
1, 42.05128205128206, 0, 0	0, 0, 0, 0
1, 43.07692307692309, 0, 0	0, 0, 0, 0
1, 44.10256410256411, 0, 0	0, 0, 0, 0
1, 45.12820512820515, 0, 0	0, 0, -1, 41.02564102564103
1, 46.15384615384618, 0, 0	0, 0, -1, 42.05128205128206
1, 47.17948717948721, 0, 0	0, 0, -1, 43.07692307692309
1, 48.20512820512824, 0, 0	0, 0, -1, 44.10256410256411
1, 49.23076923076927, 0, 0	0, 0, -1, 45.12820512820515
0, 0, 0, 0	0, 0, -1, 46.15384615384618
0, 0, 0, 0	0, 0, -1, 47.17948717948721
0, 0, 0, 0	0, 0, -1, 48.20512820512824
0, 0, 0, 0	0, 0, -1, 49.23076923076927
-1, -41.02564102564102, 0, 0	0, 0, -1, 50.25641025641029
-1, -42.05128205128204, 0, 0	0, 0, -1, 51.28205128205132
0, 0, 0, 0	0, 0, -1, 52.30769230769235

0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 1, -41.02564102564102
1, 41.02564102564103, 0, 0	0, 0, 1, -42.05128205128204
1, 42.05128205128206, 0, 0	0, 0, 1, -43.07692307692307
1, 43.07692307692309, 0, 0	0, 0, 1, -44.10256410256409
1, 44.10256410256411, -1, 41.02564102564103	0, 0, 1, -45.12820512820511
1, 45.12820512820515, -1, 42.05128205128206	0, 0, 0, 0
1, 46.15384615384618, -1, 43.07692307692309	0, 0, 0, 0
1, 47.17948717948721, 0, 0	0, 0, 0, 0
1, 48.20512820512824, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	-1, -41.02564102564102, 0, 0
0, 0, 0, 0	-1, -42.05128205128204, 0, 0
1, 41.02564102564103, 0, 0	-1, -43.07692307692307, 0, 0
1, 42.05128205128206, 0, 0	-1, -44.10256410256409, 0, 0
1, 43.07692307692309, 0, 0	-1, -45.12820512820511, -1, 41.02564102564103
1, 44.10256410256411, 0, 0	0, 0, -1, 42.05128205128206
1, 45.12820512820515, 0, 0	0, 0, -1, 43.07692307692309
0, 0, 0, 0	1, 41.02564102564103, -1, 44.10256410256411
0, 0, 0, 0	1, 42.05128205128206, -1, 45.12820512820515
0, 0, 1, -41.02564102564102	1, 43.07692307692309, -1, 46.15384615384618
0, 0, 1, -42.05128205128204	1, 44.10256410256411, -1, 47.17948717948721
0, 0, 1, -43.07692307692307	1, 45.12820512820515, -1, 48.20512820512824
0, 0, 1, -44.10256410256409	0, 0, -1, 49.23076923076927
0, 0, 1, -45.12820512820511	0, 0, -1, 50.25641025641029
0, 0, 2, -87.23346828609985	1, 41.02564102564103, 0, 0
0, 0, 2, -89.3387314439946	1, 42.05128205128206, 0, 0
0, 0, 2, -91.44399460188934	0, 0, 0, 0
0, 0, 2, -93.54925775978407	0, 0, 1, -41.02564102564102
0, 0, 1, -54.57489878542511	0, 0, 1, -42.05128205128204
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	1, 41.02564102564103, 0, 0
0, 0, -1, 41.02564102564103	1, 42.05128205128206, 0, 0
0, 0, -1, 42.05128205128206	1, 43.07692307692309, -1, 41.02564102564103
0, 0, -1, 43.07692307692309	1, 44.10256410256411, -1, 42.05128205128206
-1, -41.02564102564102, -1, 44.10256410256411	1, 45.12820512820515, -1, 43.07692307692309
-1, -42.05128205128204, 0, 0	1, 46.15384615384618, -1, 44.10256410256411
-1, -43.07692307692307, 0, 0	1, 47.17948717948721, -1, 45.12820512820515
-1, -44.10256410256409, 0, 0	1, 48.20512820512824, -1, 46.15384615384618
-1, -45.12820512820511, 0, 0	1, 49.23076923076927, -1, 47.17948717948721
-1, -46.15384615384613, 0, 0	1, 50.25641025641029, -1, 48.20512820512824
-1, -47.17948717948715, 0, 0	1, 51.28205128205132, 0, 0
-1, -48.20512820512818, 0, 0	1, 52.30769230769235, 1, -41.02564102564102

1, 53.33333333333338, 1, -42.05128205128204	0, 0, -1, 41.02564102564103
0, 0, 1, -43.07692307692307	0, 0, -1, 42.05128205128206
0, 0, 1, -44.10256410256409	1, 41.02564102564103, -2, 84.15654520917681
0, 0, 1, -45.12820512820511	1, 42.05128205128206, -2, 86.26180836707155
0, 0, 0, 0	0, 0, -1, 47.28744939271256
0, 0, 1, -41.02564102564102	0, 0, -1, 48.31309041835359
0, 0, 1, -42.05128205128204	0, 0, -1, 49.33873144399462
1, 41.02564102564103, 1, -43.07692307692307	0, 0, 0, 0
1, 42.05128205128206, 1, -44.10256410256409	0, 0, 1, -41.02564102564102
1, 43.07692307692309, 1, -45.12820512820511	0, 0, 1, -42.05128205128204
1, 44.10256410256411, 0, 0	0, 0, 1, -43.07692307692307
1, 45.12820512820515, 0, 0	0, 0, 1, -44.10256410256409
1, 46.15384615384618, 0, 0	0, 0, 1, -45.12820512820511
1, 47.17948717948721, 0, 0	0, 0, 1, -46.15384615384613
0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -41.02564102564102	0, 0, -1, 41.02564102564103
0, 0, 1, -42.05128205128204	0, 0, 0, 0
-1, -41.02564102564102, 1, -43.07692307692307	0, 0, 0, 0
-1, -42.05128205128204, 1, -44.10256410256409	0, 0, 0, 0
-1, -43.07692307692307, 1, -45.12820512820511	0, 0, 0, 0
-1, -44.10256410256409, 1, -46.15384615384613	0, 0, -1, 41.02564102564103
-1, -45.12820512820511, 1, -47.17948717948715	0, 0, -2, 83.13090418353577
-1, -46.15384615384613, 1, -48.20512820512818	0, 0, -2, 85.23616734143051
-1, -47.17948717948715, 1, -49.2307692307692	0, 0, -2, 87.34143049932525
0, 0, 0, 0	0, 0, -2, 89.44669365721998
0, 0, 0, 0	0, 0, -2, 91.55195681511472
1, 41.02564102564103, 0, 0	0, 0, 0, 0
1, 42.05128205128206, 0, 0	0, 0, 1, -41.02564102564102
1, 43.07692307692309, 0, 0	1, 41.02564102564103, 2, -83.13090418353577
0, 0, 0, 0	1, 42.05128205128206, 2, -85.23616734143051
0, 0, 1, -41.02564102564102	1, 43.07692307692309, 2, -87.34143049932525
0, 0, 1, -42.05128205128204	1, 44.10256410256411, 2, -89.44669365721998
0, 0, 1, -43.07692307692307	1, 45.12820512820515, 1, -50.47233468286102
0, 0, 1, -44.10256410256409	1, 46.15384615384618, 1, -51.49797570850204
0, 0, 1, -45.12820512820511	1, 47.17948717948721, 1, -52.52361673414306
0, 0, 1, -46.15384615384613	0, 0, 1, -53.54925775978408
-1, -41.02564102564102, 1, -47.17948717948715	0, 0, 1, -54.57489878542511
-1, -42.05128205128204, 1, -48.20512820512818	0, 0, 1, -55.60053981106613
-1, -43.07692307692307, 1, -49.2307692307692	0, 0, 1, -56.62618083670715
-1, -44.10256410256409, 1, -50.25641025641022	0, 0, -1, -15.60053981106612
-1, -45.12820512820511, 1, -51.28205128205124	0, 0, -2, 26.50472334682862
-1, -46.15384615384613, 1, -52.30769230769226	1, 41.02564102564103, -3, 69.74796659007186
-1, -47.17948717948715, 1, -53.33333333333329	1, 42.05128205128206, -3, 72.9912098333151
-1, -48.20512820512818, 1, -54.35897435897431	1, 43.07692307692309, -3, 76.23445307655834
-1, -49.2307692307692, 1, -55.38461538461533	1, 44.10256410256411, -3, 79.47769631980158
-1, -50.25641025641022, 0, 0	1, 45.12820512820515, -3, 82.72093956304483
0, 0, 0, 0	0, 0, -3, 85.96418280628807

0, 0, -3, 89.20742604953132	0, 0, -1, 43.07692307692309
0, 0, -3, 92.45066929277455	0, 0, 0, 0
0, 0, -2, 54.55593245066929	0, 0, 0, 0
0, 0, -1, 15.5815734763103	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
1, 41.02564102564103, 0, 0	0, 0, 0, 0
1, 42.05128205128206, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 1, -41.02564102564102
0, 0, 0, 0	0, 0, 1, -42.05128205128204
0, 0, 0, 0	0, 0, 1, -43.07692307692307
0, 0, 0, 0	0, 0, 1, -44.10256410256409
0, 0, 0, 0	0, 0, 1, -45.12820512820511
0, 0, 0, 0	0, 0, 1, -46.15384615384613
0, 0, 0, 0	0, 0, 1, -47.17948717948715
0, 0, 0, 0	0, 0, 1, -48.20512820512818
0, 0, 0, 0	0, 0, 0, 0
1, 41.02564102564103, 0, 0	0, 0, 0, 0
1, 42.05128205128206, 0, 0	0, 0, 0, 0
1, 43.07692307692309, 0, 0	0, 0, -1, 41.02564102564103
1, 44.10256410256411, 0, 0	0, 0, -1, 42.05128205128206
0, 0, 0, 0	0, 0, -2, 84.15654520917681
0, 0, 0, 0	0, 0, -1, 45.18218623481782
0, 0, 0, 0	0, 0, -1, 46.20782726045885
0, 0, 0, 0	0, 0, -1, 47.23346828609988
-1, -41.02564102564102, 0, 0	0, 0, -1, 48.25910931174091
-1, -42.05128205128204, 0, 0	0, 0, -1, 49.28475033738194
0, 0, 1, -41.02564102564102	1, 41.02564102564103, -1, 50.31039136302297
0, 0, 1, -42.05128205128204	1, 42.05128205128206, 0, 0
0, 0, 1, -43.07692307692307	1, 43.07692307692309, 1, -41.02564102564102
0, 0, 1, -44.10256410256409	1, 44.10256410256411, 1, -42.05128205128204
0, 0, 1, -45.12820512820511	1, 45.12820512820515, 0, 0
0, 0, 2, -87.23346828609985	1, 46.15384615384618, 0, 0
0, 0, 2, -89.3387314439946	1, 47.17948717948721, 0, 0
0, 0, 2, -91.44399460188934	1, 48.20512820512824, 0, 0
0, 0, 2, -93.54925775978407	0, 0, 0, 0
0, 0, 2, -95.65452091767881	0, 0, 0, 0
-1, -41.02564102564102, 1, -56.68016194331984	0, 0, 0, 0
-1, -42.05128205128204, 1, -57.70580296896087	0, 0, -1, 41.02564102564103
-1, -43.07692307692307, 1, -58.73144399460189	1, 41.02564102564103, -1, 42.05128205128206
-1, -44.10256410256409, 1, -59.75708502024291	1, 42.05128205128206, -1, 43.07692307692309
-1, -45.12820512820511, 1, -60.78272604588394	1, 43.07692307692309, -1, 44.10256410256411
-1, -46.15384615384613, -1, -19.75708502024291	1, 44.10256410256411, -1, 45.12820512820515
-1, -47.17948717948715, 0, 0	1, 45.12820512820515, -1, 46.15384615384618
0, 0, 0, 0	1, 46.15384615384618, -1, 47.17948717948721
0, 0, 0, 0	1, 47.17948717948721, 0, 0
0, 0, -1, 41.02564102564103	1, 48.20512820512824, 0, 0
0, 0, -1, 42.05128205128206	1, 49.23076923076927, 0, 0



1, 50.25641025641029, 0, 0	0, 0, 1, -49.2307692307692
0, 0, 0, 0	0, 0, 1, -50.25641025641022
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, -1, 41.02564102564103
-1, -41.02564102564102, 0, 0	0, 0, -1, 42.05128205128206
0, 0, 0, 0	0, 0, -1, 43.07692307692309
0, 0, 0, 0	0, 0, -1, 44.10256410256411
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	-1, -41.02564102564102, 0, 0
0, 0, 0, 0	-1, -42.05128205128204, 0, 0
0, 0, 0, 0	-1, -43.07692307692307, 0, 0
1, 41.02564102564103, 0, 0	-1, -44.10256410256409, -1, 41.02564102564103
1, 42.05128205128206, 0, 0	0, 0, -1, 42.05128205128206
1, 43.07692307692309, 0, 0	1, 41.02564102564103, -1, 43.07692307692309
1, 44.10256410256411, 0, 0	1, 42.05128205128206, -1, 44.10256410256411
1, 45.12820512820515, 0, 0	1, 43.07692307692309, -1, 45.12820512820515
0, 0, 0, 0	2, 85.18218623481784, -1, 46.15384615384618
0, 0, 0, 0	2, 87.28744939271257, -1, 47.17948717948721
-1, -41.02564102564102, 0, 0	2, 89.39271255060731, 0, 0
-1, -42.05128205128204, 0, 0	1, 50.41835357624834, 0, 0
0, 0, 1, -41.02564102564102	1, 51.44399460188937, 0, 0
0, 0, 1, -42.05128205128204	1, 52.4696356275304, 0, 0
-1, -41.02564102564102, 1, -43.07692307692307	0, 0, 0, 0
-1, -42.05128205128204, 1, -44.10256410256409	0, 0, 0, 0
0, 0, 2, -86.20782726045883	0, 0, 0, 0
0, 0, 2, -88.31309041835357	-1, -41.02564102564102, -1, 41.02564102564103
0, 0, 2, -90.41835357624832	-1, -42.05128205128204, -1, 42.05128205128206
0, 0, 2, -92.52361673414304	-1, -43.07692307692307, -1, 43.07692307692309
0, 0, 2, -94.62887989203779	0, 0, -1, 44.10256410256411
0, 0, 2, -96.73414304993253	1, 41.02564102564103, -1, 45.12820512820515
-1, -41.02564102564102, 1, -57.75978407557356	1, 42.05128205128206, -1, 46.15384615384618
-1, -42.05128205128204, 0, 0	1, 43.07692307692309, -1, 47.17948717948721
-1, -43.07692307692307, 0, 0	1, 44.10256410256411, -1, 48.20512820512824
-1, -44.10256410256409, -1, 41.02564102564103	1, 45.12820512820515, 0, 0
-1, -45.12820512820511, -1, 42.05128205128206	1, 46.15384615384618, 0, 0
-1, -46.15384615384613, -1, 43.07692307692309	1, 47.17948717948721, 1, -41.02564102564102
0, 0, 0, 0	1, 48.20512820512824, 1, -42.05128205128204
0, 0, 1, -41.02564102564102	1, 49.23076923076927, 1, -43.07692307692307
1, 41.02564102564103, 1, -42.05128205128204	0, 0, 1, -44.10256410256409
0, 0, 1, -43.07692307692307	0, 0, 0, 0
0, 0, 1, -44.10256410256409	0, 0, 0, 0
0, 0, 1, -45.12820512820511	0, 0, 0, 0
0, 0, 1, -46.15384615384613	0, 0, 0, 0
0, 0, 1, -47.17948717948715	0, 0, 0, 0
0, 0, 1, -48.20512820512818	1, 41.02564102564103, 0, 0

1, 42.05128205128206, 0, 0	0, 0, 1, -46.15384615384613
1, 43.07692307692309, 0, 0	0, 0, 0, 0
1, 44.10256410256411, 0, 0	0, 0, 0, 0
1, 45.12820512820515, 1, -41.02564102564102	0, 0, 0, 0
1, 46.15384615384618, 1, -42.05128205128204	0, 0, 0, 0
1, 47.17948717948721, 1, -43.07692307692307	0, 0, 0, 0
0, 0, 1, -44.10256410256409	0, 0, 0, 0
0, 0, 1, -45.12820512820511	0, 0, 0, 0
0, 0, 1, -46.15384615384613	0, 0, 0, 0
-1, -41.02564102564102, 1, -47.17948717948715	0, 0, 0, 0
0, 0, 1, -48.20512820512818	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -41.02564102564102	0, 0, 0, 0
0, 0, 1, -42.05128205128204	0, 0, 0, 0
0, 0, 1, -43.07692307692307	0, 0, -1, 41.02564102564103
0, 0, 1, -44.10256410256409	0, 0, -1, 42.05128205128206
0, 0, 1, -45.12820512820511	0, 0, -1, 43.07692307692309
0, 0, 1, -46.15384615384613	0, 0, -1, 44.10256410256411
0, 0, 1, -47.17948717948715	0, 0, -1, 45.12820512820515
0, 0, 1, -48.20512820512818	0, 0, -1, 46.15384615384618
0, 0, 1, -49.2307692307692	0, 0, -1, 47.17948717948721
0, 0, 1, -50.25641025641022	0, 0, -1, 48.20512820512824
0, 0, 1, -51.28205128205124	0, 0, -1, 49.23076923076927
0, 0, 1, -52.30769230769226	0, 0, 0, 0
0, 0, 0, 0	0, 0, 1, -41.02564102564102
0, 0, 0, 0	1, 41.02564102564103, 1, -42.05128205128204
0, 0, 0, 0	1, 42.05128205128206, 1, -43.07692307692307
0, 0, 0, 0	1, 43.07692307692309, 1, -44.10256410256409
0, 0, 0, 0	1, 44.10256410256411, 1, -45.12820512820511
0, 0, -1, 41.02564102564103	1, 45.12820512820515, 1, -46.15384615384613
0, 0, 0, 0	1, 46.15384615384618, 1, -47.17948717948715
0, 0, 0, 0	1, 47.17948717948721, 1, -48.20512820512818
0, 0, 0, 0	0, 0, 1, -49.2307692307692
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	-1, -41.02564102564102, 0, 0
0, 0, 0, 0	-1, -42.05128205128204, -1, 41.02564102564103
0, 0, 0, 0	-1, -43.07692307692307, -1, 42.05128205128206
0, 0, 1, -41.02564102564102	-1, -44.10256410256409, -2, 84.15654520917681
0, 0, 1, -42.05128205128204	0, 0, -1, 45.18218623481782
0, 0, 1, -43.07692307692307	0, 0, 0, 0
0, 0, 1, -44.10256410256409	0, 0, 0, 0
0, 0, 1, -45.12820512820511	0, 0, -1, 41.02564102564103

1, 41.02564102564103, 0, 0	1, 45.12820512820515, 0, 0	0, 0, 1, -43.07692307692307
1, 42.05128205128206, 1, -41.02564102564102	1, 46.15384615384618, 0, 0	0, 0, 1, -44.10256410256409
1, 43.07692307692309, 1, -42.05128205128204	1, 47.17948717948721, 0, 0	0, 0, 1, -45.12820512820511
0, 0, 1, -43.07692307692307	1, 48.20512820512824, 0, 0	0, 0, 1, -46.15384615384613
0, 0, 1, -44.10256410256409	1, 49.23076923076927, 0, 0	0, 0, 0, 0
0, 0, 1, -45.12820512820511	0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -46.15384615384613	0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -47.17948717948715	-1, -41.02564102564102, 0, 0	0, 0, 0, 0
0, 0, 1, -48.20512820512818	-1, -42.05128205128204, 0, 0	0, 0, -1, 41.02564102564103
0, 0, 1, -49.2307692307692	-1, -43.07692307692307, 0, 0	0, 0, -1, 42.05128205128206
0, 0, 1, -50.25641025641022	0, 0, 0, 0	-1, -41.02564102564102, 0, 0
1, 41.02564102564103, 1, -51.28205128205124	0, 0, 0, 0	-1, -42.05128205128204, 0, 0
1, 42.05128205128206, 0, 0	0, 0, 0, 0	0, 0, 0, 0
1, 43.07692307692309, 0, 0	0, 0, 0, 0	0, 0, 0, 0
0, 0, -1, 41.02564102564103	0, 0, 0, 0	0, 0, 0, 0
0, 0, -1, 42.05128205128206	0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -41.02564102564102	0, 0, 0, 0	0, 0, 0, 0
1, 41.02564102564103, 1, -42.05128205128204	0, 0, 0, 0	0, 0, 0, 0
1, 42.05128205128206, 1, -43.07692307692307	0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -44.10256410256409	0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -45.12820512820511	0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -46.15384615384613	0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -47.17948717948715	0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -48.20512820512818	0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0
1, 41.02564102564103, 0, 0	0, 0, 0, 0	0, 0, 0, 0
1, 42.05128205128206, 0, 0	0, 0, 0, 0	0, 0, 0, 0
1, 43.07692307692309, 0, 0	0, 0, 0, 0	0, 0, 0, 0
1, 44.10256410256411, 0, 0	0, 0, 0, 0	0, 0, 0, 0
1, 45.12820512820515, 0, 0	0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	1, 41.02564102564103, 0, 0	0, 0, 0, 0
0, 0, 0, 0	1, 42.05128205128206, 0, 0	0, 0, 0, 0
0, 0, 0, 0	1, 43.07692307692309, 0, 0	0, 0, 0, 0
-1, -41.02564102564102, 0, 0	1, 44.10256410256411, 0, 0	0, 0, 0, 0
-1, -42.05128205128204, 0, 0	1, 45.12820512820515, 0, 0	0, 0, 0, 0
0, 0, 0, 0	1, 46.15384615384618, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0
1, 41.02564102564103, 0, 0	-1, -41.02564102564102, 0, 0	0, 0, 0, 0
1, 42.05128205128206, 0, 0	-1, -42.05128205128204, 0, 0	0, 0, 0, 0
1, 43.07692307692309, 0, 0	0, 0, 1, -41.02564102564102	0, 0, 0, 0
1, 44.10256410256411, 0, 0	0, 0, 1, -42.05128205128204	0, 0, 0, 0

0, 0, 0, 0	0, 0, 1, -43.07692307692307
0, 0, 0, 0	0, 0, 1, -44.10256410256409
0, 0, 0, 0	0, 0, 0, 0
1, 41.02564102564103, 0, 0	0, 0, 0, 0
1, 42.05128205128206, 0, 0	0, 0, 0, 0
1, 43.07692307692309, 0, 0	0, 0, 0, 0
1, 44.10256410256411, -1, 41.02564102564103	0, 0, 0, 0
1, 45.12820512820515, -1, 42.05128205128206	0, 0, 0, 0
1, 46.15384615384618, -1, 43.07692307692309	0, 0, 0, 0
0, 0, -1, 44.10256410256411	0, 0, -1, 41.02564102564103
0, 0, -1, 45.12820512820515	0, 0, 0, 0
0, 0, -1, 46.15384615384618	1, 41.02564102564103, 0, 0
-1, -41.02564102564102, -1, 47.17948717948721	1, 42.05128205128206, 0, 0
-1, -42.05128205128204, -1, 48.20512820512824	1, 43.07692307692309, 0, 0
0, 0, 0, 0	1, 44.10256410256411, 0, 0
0, 0, 1, -41.02564102564102	1, 45.12820512820515, 0, 0
1, 41.02564102564103, 2, -83.13090418353577	1, 46.15384615384618, 0, 0
1, 42.05128205128206, 2, -85.23616734143051	0, 0, 0, 0
0, 0, 2, -87.34143049932525	0, 0, 0, 0
0, 0, 1, -48.36707152496627	0, 0, 0, 0
0, 0, 1, -49.39271255060731	-1, -41.02564102564102, 0, 0
0, 0, 1, -50.41835357624833	-1, -42.05128205128204, 0, 0
1, 41.02564102564103, 1, -51.44399460188935	-1, -43.07692307692307, 0, 0
1, 42.05128205128206, 1, -52.46963562753037	0, 0, 0, 0
1, 43.07692307692309, 0, 0	0, 0, 0, 0
1, 44.10256410256411, 0, 0	0, 0, 0, 0
1, 45.12820512820515, 0, 0	0, 0, 1, -41.02564102564102
1, 46.15384615384618, 0, 0	0, 0, 1, -42.05128205128204
1, 47.17948717948721, 0, 0	0, 0, 1, -43.07692307692307
0, 0, -2, 42.10526315789474	0, 0, 1, -44.10256410256409
0, 0, -1, 3.130904183535762	0, 0, 1, -45.12820512820511
0, 0, -1, 4.156545209176787	0, 0, 1, -46.15384615384613
-1, -41.02564102564102, -1, 5.182186234817813	0, 0, 1, -47.17948717948715
-1, -42.05128205128204, -1, 6.207827260458838	0, 0, 1, -48.20512820512818
0, 0, -1, 7.233468286099864	0, 0, 0, 0
0, 0, -2, 49.33873144399461	0, 0, 0, 0
0, 0, -2, 51.44399460188935	0, 0, 0, 0
0, 0, -2, 53.54925775978408	0, 0, 0, 0
0, 0, -2, 55.65452091767883	0, 0, 0, 0
0, 0, -2, 57.75978407557357	0, 0, 0, 0
0, 0, -2, 59.8650472334683	0, 0, 0, 0
0, 0, -1, 20.89068825910933	0, 0, 0, 0
0, 0, -1, 21.91632928475035	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 1, -41.02564102564102	0, 0, 0, 0
0, 0, 1, -42.05128205128204	0, 0, 0, 0



0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	-1, -41.02564102564102, 0, 0
0, 0, 0, 0	-1, -42.05128205128204, -2, 42.10526315789474
0, 0, 0, 0	0, 0, -1, 3.130904183535762
0, 0, 0, 0	1, 41.02564102564103, -1, 4.156545209176787
0, 0, 0, 0	0, 0, -1, 5.182186234817813
0, 0, 0, 0	0, 0, -2, 47.28744939271255
0, 0, 0, 0	0, 0, -2, 49.3927125506073
0, 0, 0, 0	0, 0, -2, 51.49797570850204
0, 0, 0, 0	1, 41.02564102564103, -2, 53.60323886639677
0, 0, 0, 0	1, 42.05128205128206, -2, 55.70850202429152
0, 0, 0, 0	1, 43.07692307692309, -2, 57.81376518218625
0, 0, 0, 0	1, 44.10256410256411, -1, 18.83940620782727
0, 0, 0, 0	1, 45.12820512820515, 0, 0
0, 0, 0, 0	1, 46.15384615384618, 0, 0
0, 0, 0, 0	1, 47.17948717948721, 1, -41.02564102564102
0, 0, 0, 0	1, 48.20512820512824, 1, -42.05128205128204
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	-1, -41.02564102564102, 0, 0
0, 0, 0, 0	-1, -42.05128205128204, -1, 41.02564102564103
0, 0, 0, 0	0, 0, 0, 0
0, 0, 0, 0	0, 0, 0, 0
0, 0, -1, 41.02564102564103	1, 41.02564102564103, 0, 0
0, 0, -1, 42.05128205128206	1, 42.05128205128206, 0, 0
0, 0, -1, 43.07692307692309	1, 43.07692307692309, -1, 41.02564102564103
0, 0, -1, 44.10256410256411	0, 0, -1, 42.05128205128206
0, 0, -1, 45.12820512820515	0, 0, -1, 43.07692307692309
0, 0, -1, 46.15384615384618	0, 0, -1, 44.10256410256411
0, 0, -2, 88.25910931174091	1, 41.02564102564103, 0, 0
0, 0, -2, 90.36437246963565	1, 42.05128205128206, 0, 0
0, 0, -2, 92.4696356275304	1, 43.07692307692309, 0, 0
0, 0, 0, 0	1, 44.10256410256411, 1, -41.02564102564102
0, 0, 1, -41.02564102564102	1, 45.12820512820515, 1, -42.05128205128204
0, 0, 2, -83.13090418353577	1, 46.15384615384618, 1, -43.07692307692307
0, 0, 3, -100	1, 47.17948717948721, 1, -44.10256410256409
1, 41.02564102564103, 3, -100	1, 48.20512820512824, 0, 0
1, 42.05128205128206, 3, -100	1, 49.23076923076927, 0, 0
1, 43.07692307692309, 3, -100	0, 0, 0, 0
1, 44.10256410256411, 3, -100	0, 0, 0, 0
1, 45.12820512820515, 3, -100	0, 0, 1, -41.02564102564102
1, 46.15384615384618, 2, -62.10526315789473	-1, -41.02564102564102, 1, -42.05128205128204
1, 47.17948717948721, 2, -64.21052631578947	-1, -42.05128205128204, 1, -43.07692307692307
1, 48.20512820512824, 0, 0	0, 0, 1, -44.10256410256409
0, 0, 0, 0	0, 0, 1, -45.12820512820511
0, 0, 0, 0	1, 41.02564102564103, 0, 0

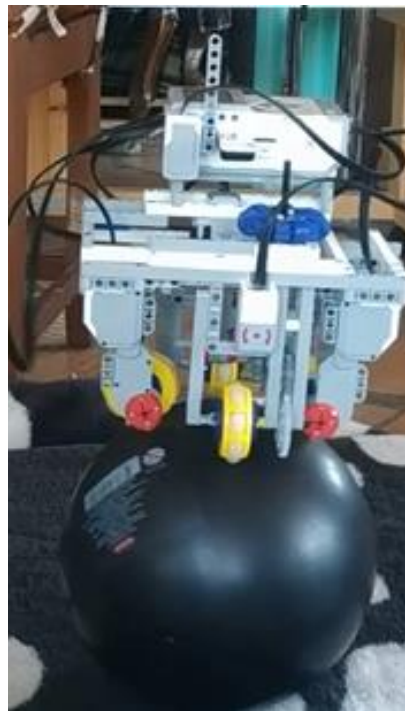
1, 42.05128205128206, 0, 0	
1, 43.07692307692309, 0, 0	
1, 44.10256410256411, 0, 0	
1, 45.12820512820515, 1, -41.02564102564102	
1, 46.15384615384618, 1, -42.05128205128204	
1, 47.17948717948721, 1, -43.07692307692307	
0, 0, 1, -44.10256410256409	
-1, -41.02564102564102, 2, -86.20782726045883	
-1, -42.05128205128204, 2, -88.31309041835357	
-1, -43.07692307692307, 2, -90.41835357624832	
0, 0, 1, -51.44399460188934	
0, 0, 1, -52.46963562753037	
1, 41.02564102564103, 0, 0	
1, 42.05128205128206, 0, 0	
1, 43.07692307692309, -1, 41.02564102564103	
1, 44.10256410256411, -1, 42.05128205128206	
2, 86.20782726045887, 0, 0	
2, 88.31309041835359, 0, 0	
1, 49.33873144399462, 0, 0	
1, 50.36437246963565, 0, 0	
1, 51.39001349527668, 0, 0	
0, 0, 0, 0	
0, 0, 0, 0	
0, 0, 0, 0	
0, 0, 0, 0	
1, 41.02564102564103, 1, -41.02564102564102	
2, 83.13090418353577, 1, -42.05128205128204	
3, 100, 1, -43.07692307692307	
3, 100, 1, -44.10256410256409	
3, 100, 1, -45.12820512820511	
2, 62.10526315789473, 2, -87.23346828609985	
2, 64.21052631578947, 1, -48.25910931174088	
1, 25.23616734143049, 1, -49.28475033738192	
1, 26.26180836707151, 1, -50.31039136302294	
-1, -14.76383265856951, 1, -51.33603238866396	
-3, -58.00707590181275, -1, -10.31039136302292	
-3, -61.250319145056, -1, -9.284750337381897	
-3, -64.49356238829925, -1, -8.259109311740872	
-4, -100, -1, -7.233468286099846	
-4, -100, -1, -6.207827260458821	
-4, -100, -1, -5.182186234817795	
-3, -63.24324324324325, -1, -4.15654520917677	
-2, -25.34850640113798, 0, 0	
-1, 13.625852573221, 0, 0	
-1, 12.60021154757997, 0, 0	
0, 0, 0, 0	
0, 0, 1, -41.02564102564102	
-1, -41.02564102564102, 1, -42.05128205128204	
-1, -42.05128205128204, 1, -43.07692307692307	
-1, -43.07692307692307, 1, -44.10256410256409	
-1, -44.10256410256409, 1, -45.12820512820511	
0, 0, 2, -87.23346828609985	
0, 0, 2, -89.3387314439946	
1, 41.02564102564103, 2, -91.44399460188934	
0, 0, 2, -93.54925775978407	
<b>0, 0, 1, -54.57489878542511</b>	

#### Anexo 6. Funcionamiento del Ballbot EV3

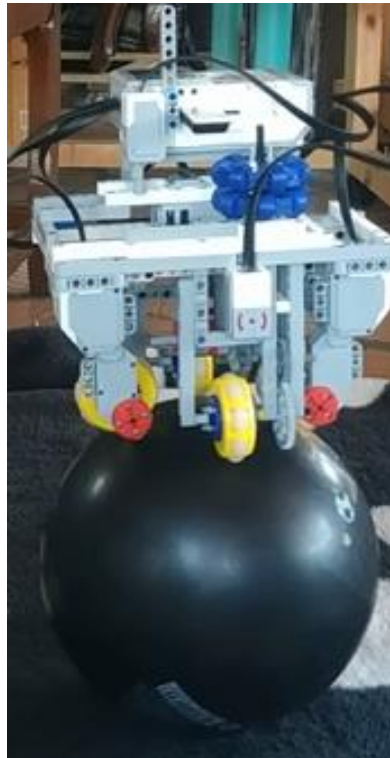
- Sin peso, plano inclinado de  $3^{\circ}$  con respecto a la horizontal



- Con peso adicional (45 gr), plano inclinado de  $3^{\circ}$  con respecto a la horizontal



- Con peso adicional (90 gr), plano inclinado de  $3^\circ$  con respecto a la horizontal

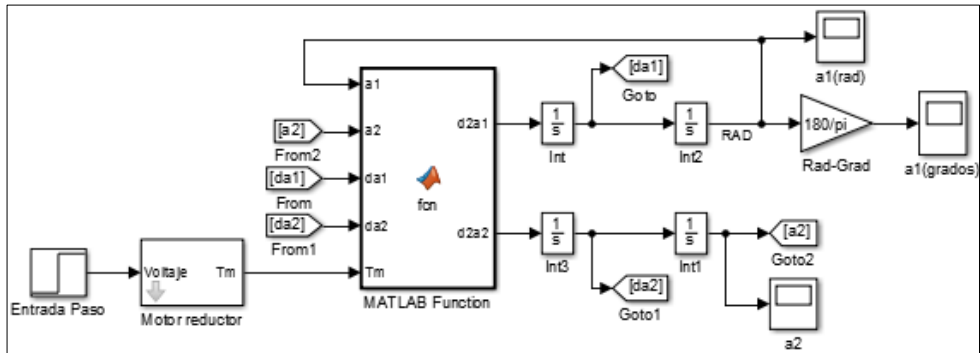


- Con peso adicional (135 gr), plano inclinado de  $3^\circ$  con respecto a la horizontal



## Anexo 7. Sintonización PID Ziegler Nichols en Lazo Abierto

Se coloca al sistema del robot en lazo abierto para realizar la sintonización ante una entrada paso unitario.



En la Figura siguiente se presenta la salida del sistema cuya forma de onda se asemeja a una función exponencial, por tanto, no posee una curva en forma de  $S$  y no permite aplicar el primer método de sintonización por Ziegler Nichols.

